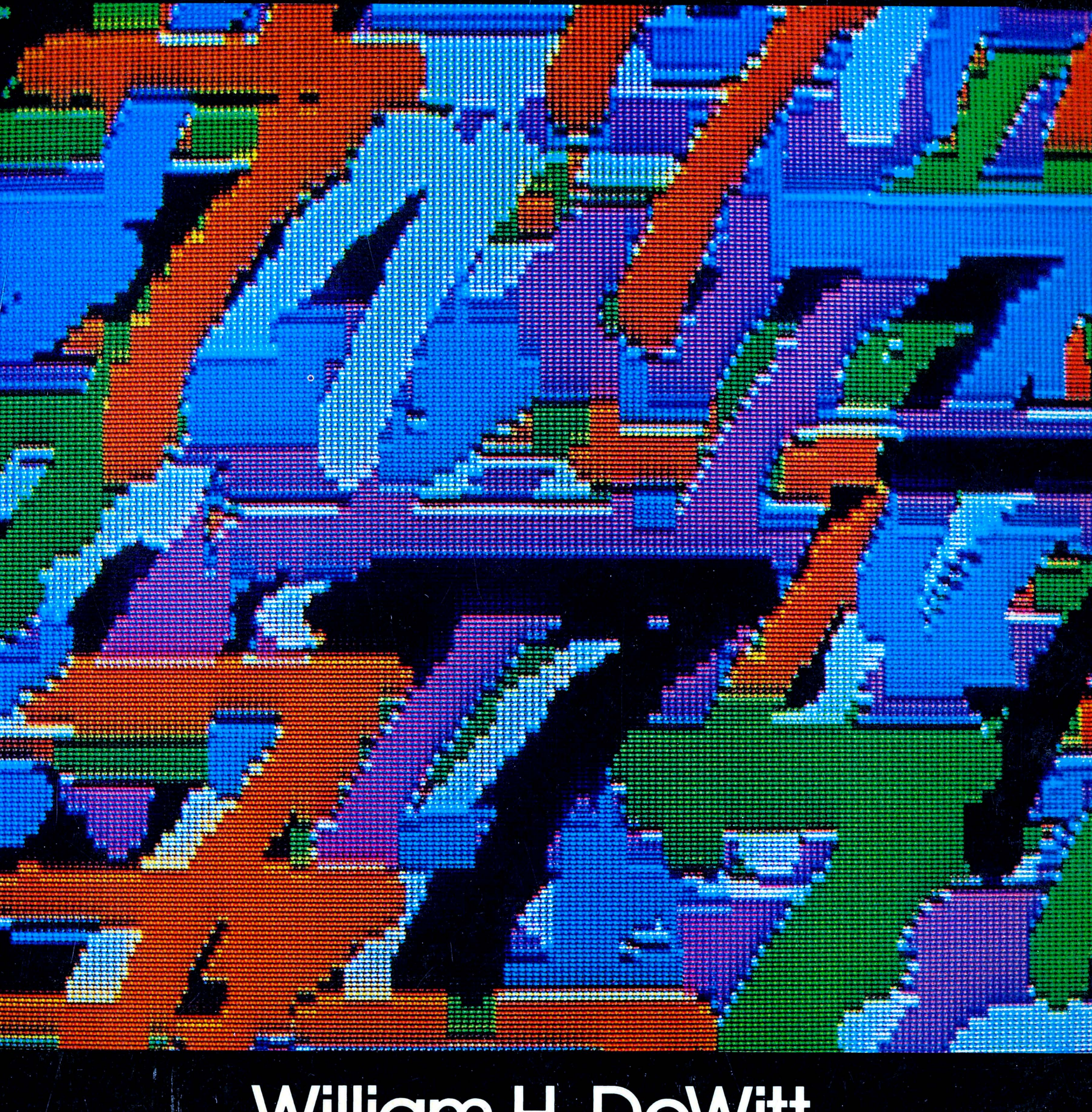
HiRes/Double HiRes
GRAPHICS

For the Apple IIc
and Apple II Family



William H. DeWitt

HiRes/Double HiRes Graphics for the Apple IIc and Apple II Family

William H. DeWitt

A Wiley Press Book

JOHN WILEY & SONS, INC.
New York • Chichester • Brisbane • Toronto • Singapore

Publisher: Stephen Kippur
Editor: Theron Shreve
Managing Editor: Katherine Schowalter
Design, Composition & Make-Up: G&H/SOHO, Ltd.

Copyright © 1986 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Apple® is a registered trademark of Apple Computer, Inc.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging-in-Publication Data

DeWitt, William H.
HiRes/Double HiRes graphics for the Apple IIc
and Apple II family.

Includes index.
1. Apple IIc (Computer)—Programming. 2. Apple
IIe (Computer)—Programming. 3. Computer graphics.
I. Title.
QA76.8.A662D49 1986 006.6′76 85-26018
ISBN 0-471-83183-2

Printed in the United States of America

86 87 10 9 8 7 6 5 4 3 2 1

Wiley Press books can be used as premiums to promote products or services, for training, or in mail-order catalogs. For information on quantity discounts, please write to the Special Sales Department, John Wiley & Sons, Inc.

Contents

Introduction x
Acknowledgments xiii

SECTION 1 The Programming Path to Low and HiRes Graphics 1

1. **The Facts of Life with a Computer** 3

A DOS 3.3 Overview □ A ProDOS Overview □ Summary □ Making a Choice □ Get Organized □ Be Economical □ Don't Be Intimidated □ Pay Attention to Signs and Sequences □ Some Programming Tools □ Random Numbers □ A Notable Exception to Remember

2. **Low-Resolution Graphics** 11

The Low-Resolution Screen □ GR Commands and Colors □ GR Commands at Work □ Text and Images □ Using the Entire GR Screen □ Where Do We Go from Here? □ Low Can Be Plus or Minus □ Color Calibration □ Plotting Dots □ Next Stop on the HLIN, VLIN □ "Blocking In" Areas □ Filling the Entire Screen with One Color □ Togetherness Really Works □ Curves Ahead □ More Low Resolution Later □ Conceptual Programming □ Onward and Upward to High-Resolution Graphics

3. **High-Resolution Graphics** 23

A Whole New World □ Just Like Television □ High-Resolution Commands □ HGR Colors □ A Three-Way Command □ To Plot a Dot □ The HPLOT Thickens □ Another HPLOT Uncovered □ Try It Yourself □ Demonstration Programs □ The Art of Curves □ A Keep-It-Simple Circle □ A Calculated Look □ HPLOTting into Space □ Want to Double-Up?

4. **Getting into Shape** 33

Shape Defined □ Shape Commands □ About Shapes and Shape Tables □ Protocol Forever □ Apple Bytes Shape Data □ In Shape at Last □ Shape Table Details □ Rounding the Square □ Shape Up or Ship Out? □ Carousel, Anyone? □ Shape Tables Galore □ Getting Shapes Together □ Getting the Picture □ Demonstrating Shapes □ Triangular Geometrics □ Corridorically Speaking □ Spacewalking Shapes

5. **Animation** 55

Some Common Ground □ Apple Picking □ Long May She Wave □ Casso Comes to Life □ Horticultural Happenings □ Plantin' Time □ Animated Conversations □ Talk Lines □ Make Casso a Flasher □ Traveling Is Broadening

6. **Creating a Slide Show** 65

Easy Does It □ Get an Album □ Pix in the Fridge □ The Checkout Counter □ Pictures Galore □ Getting Control □ The Real Thing □ About A$4000 □ Photographing Your Monitor Screen □ Seeing the Subject Objectively □ Ground Rules □ Go the Professional Route □ The 35-mm Approach □ Going the Instant Route □ Serendipity 35 □ The Preliminaries □ The Nitty-Gritty □ You Press the Button □ A Filter Tip □ Further Information Department □ Lens Attachment Distributors □ Videotaping □ The Line-Up □ Video □ What VCR? □ Getting It Together □ Wait One! □ The Objective Is . . .

□ Nobody's Perfect □ Some Practical Instructions □ Computergate □ Ye Editor □ The Not-Too-Candid Camera □ On the Plus Side □ Give or Take □ Camera Action □ Don't Just Stand There! □ Sound On □ Rhythm and Blues □ Be a Believer □ Turning "Pro" □ A Point of Departure

SECTION 2 Choosing and Using Graphics Software 89

7. **Introducing Software Terminology and Double HiRes** 91

Software Talk □ System Disks □ Configuring the System Disk □ The Nitty-Gritty of Configuration □ Font Disks □ Data Disks □ Graphics Input Devices □ How Do They Do It? □ Menu-Driven Systems □ Windows □ Data Disk Details □ Preparing a Data Disk for DOS 3.3 Operation □ Preparing a Data Disk for ProDOS Operation □ Double HiRes—A New Experience □ The Ampersand Commands □ Quotes and .AUX □ What Makes Double HiRes?

8. **Picture It with Dazzle Draw** 99

What Is It? □ It's Versatile and Understandable □ What Can I Do with Dazzle Draw? □ Setting Up the Dazzle Draw System Disk □ The Mouse Does All the Work □ Putting Dazzle Draw to Work □ What's Needed □ A Minor Assumption □ It's Mouse Aerobic Time □ You Need a Data Disk to Store Your Pictures □ Design a General Form for Cards or Letterheads □ Project Complete □ More Dazzle Draw Pix □ Dazzle to Go?

9. **Fontrix Is Your Type** 109

What Is It? □ Graphics Input Devices □ Which Tricks with Fontrix? □ More About the Fontrix Package □ It's Fully Packed □ Menu Time □ About the User's Manual □ Add a Title to a Picture □ What's Needed □ Starting Up □ Save Composition VII?

10. **Everyman's Print Shop** 119

What's a Print Shop? □ Print Anything—Well, Practically Anything! □ Remember George Eastman? □ The Manual's Good, Too □ Setting Up the Print Shop System Disk □ A la Carte □ Outstanding in the Field □ What's Needed To Use Print Shop □ The System Disk □ About Data Disks □ It's Fast □ Design a Book Cover □ The Design Steps □ More!

11. **Beagle Graphics** 129

What Is It? □ Hardware Requirements □ Pointing or Graphics Input Devices □ Printing Your Graphics Creations □ Details You Wouldn't Want to Miss □ The Menus □ A Solid Gold Paperback? □ Beagle Graphics at Work □ Create a Poster or Fabric Design □ What's Needed for this Project? □ Getting Started □ The Finale

12. **Conversion and Other Conveniences** 139

Yin and Yang □ About Picture Files □ Sounds Complicated? It Isn't! □ Beagle to Dazzle and Vice Versa □ First, Beagle to Dazzle □ Dazzle to Beagle, Over! □ For Complete Graphics System Fans Only □ Converting HiRes Pictures to Double HiRes □ Saving Double HiRes Pictures □ Displaying Double HiRes Pictures □ The End Is Near

13. **Programming in Double HiRes** 147

Doubles, Anyone? □ An Acknowledgment □ Three Important Reminders □ A Time-Saving Suggestion □ Startup □ Easy Does It □ A Program Conversion, HGR to DHGR □ Now Compare the Old and the New □ After "Ampersanding" □ Custom Finish □ All That Glitters □ Programming in DHGR from Scratch □ Double HiRes from the Start

14. **Fantavision** 157

What Is Fantavision? □ An Important Note □ What's Needed for Fantavision? □ Movies a la Fantavision □ The Fantavision Concept □ Fantavision Features □ A Fantavision Project □ Looking Ahead

Appendix: More Demonstration Programs 165

For the Finale □ Speed and Color □ The Last Sunset □ Copycat Styling □ Strike Up the Band! □ Coming Abstractions □ Cascading Color □ Computempo, Anyone? □ In the Best Circles

Index 175

List of Programs

2.1 Drawing Lines with HLIN, VLIN 12
2.2 Adding Text to Graphics 13
2.3 Apple Color Bars 15
2.4 Random Dot Plots 16
2.5 Framed Internality 17
2.6 Blocking in an Area 18
2.7 Screen Color 18
2.8 Flowchart 19
2.9 A Curved Line 20
3.1 Starry Sky 25
3.2 Screen Border 26
3.3 The Art of Curves 28
3.4 A Simple Circle 29
3.5 Tubular Dynamics 30
3.6 Pulsar 31
4.1 A Square Shape 37
4.2 Rounding the Square 40
4.3 The Shape Tester 42
4.4 Rotational Shape Test 43
4.5 Triangular Geometrics 48
4.6 Doorless Corridor 49
4.7 Another Corridor 49
4.8 Magic Wands 50
4.9 Rotating Plane 51

4.10 **Dual Rotating Planes** 52
4.11 **Crosses Crossing** 53
5.1 **Old Compuglory Ripples** 56
5.2 **The Incomparable Casso Talks** 59
5.3 **Butterflies and Blossoms** 62
5.4 **Traveling Text** 63
6.1 **Slide Show #1** 68
12.1 **Convert Beagle Graphics to Dazzle Draw** 141
12.2 **Convert Dazzle Draw to Beagle Graphics** 142
12.3 **Convert Dazzle Draw to Complete Graphics System** 143
12.4 **Convert HiRes Pictures to Double HiRes and Save** 145
12.5 **Save DHR Pictures** 145
12.6 **Display DHR Pictures** 146
13.1 **Startup** 149
13.2 **Pulsar** 151
13.3 **DHR Pulsar** 152
13.4 **Final DHR Pulsar** 153
13.5 **Ampersand Demo** 156
A1.1 **Fastrack Colors** 166
A1.2 **Sunburst** 166
A1.3 **Mondrianish** 167
A1.4 **Musique Triangles** 169
A1.5 **Triangular Abstractions** 170
A1.6 **Color Cascade** 170
A1.7 **Computempo** 171
A1.8 **Tempo Dots** 172
A1.9 **Shapecircle** 172

Introduction

The main purpose of this book is to familiarize you with the graphics capabilities of the Apple computers and to demonstrate how to use them for your own artistic or design purposes.

You can "paint" pictures or "draw" designs with your Apple IIc/IIe computer by programming or by the use of software/hardware packages. This book will help you decide which of these two paths to follow.

Chapters 1 through 6 contain helpful introductory information and programs that you can use to create graphics images in the original Apple low- and high-resolution modes.

These early chapters contain important programming and operational details, explanations of low- and high-resolution commands, a discussion of shapes and shape tables, and dozens of demonstration programs. Also included are: programs to display shapes prior to use in a program; suggestions for conceptual programming; and instructions for displaying your pictures in slide-show form using the computer, video, or photography.

The images created by these programs and others throughout the book are dynamic, changing in color and configuration as the program proceeds. The rate at which image changes occur adds a fascinating dimension to the presentation on your monitor screen. (Static images can be created by stopping the program action at preselected points or as desired image/color combinations develop.) I urge you to run these programs and experiment with them to see how applying your imagination can sometimes create effects entirely different from those of the original program.

The original graphics capabilities of the Apple II series computers have been enhanced by the availability of "Double HiRes," a software approach to increasing the number of HiRes colors and doubling picture resolution. Double HiRes also offers dozens of other exciting image-forming features.

There are easy-to-follow hands-on explanations of double HiRes and some of the best graphics packages using this system in Chapters 7 through 13. Chapter 14 introduces the new software animation package "Fantavision."

The programs of Chapters 2, 3, 4, 5, and 13 demonstrate how lively programmed graphics can be. However, if you prefer drawing on the screen to programming, you should go the graphics package route and put your artistic talents to work. The disk operating systems and software terminology used with graphics packages are discussed in Chapter 7.

Graphics packages generally include a disk containing the software necessary to control the drawing features of the package, and an instruction manual. Many also include some form of graphics tablet with a light pen or other device for drawing. This means that by using one of these graphics packages, you can easily draw screen images in much the same manner as with a pencil or brush. "Pictures" created in this fashion are more like conventional drawings or paintings than are programmed images, since they are static in nature and will remain wherever they are drawn on the screen. However, some software packages can be used to create a series of images differing in shape or position for animation purposes.

Buying several graphics packages in order to find out how they work and which one suits your needs can be an expensive (and sometimes frustrating) experience. To give you some basis for evaluating graphics packages before you spend your hard-earned money, Chapters 8 through 13 of this book will introduce you to three excellent software packages and show you the capabilities you can gain by using them. You'll enjoy using these graphics aids when you see how creatively they can be applied with the Apple IIc/IIe computers.

I hope that this book will help you make effective use of the programming and software approaches to creating images. Whichever route you choose, remember that the key to learning how to make the best use of your Apple's graphics capabilities is to experiment!

WILLIAM H. DEWITT

Acknowledgments

I acknowledge with thanks permission to describe the use of the software packages, "Dazzle Draw," "The Print Shop," and "Fantavision," including their user manuals, published by Broderbund Software, Inc. "Dazzle Draw" was written by David Snider, copyright 1984 by Broderbund Software. The design and development of "The Print Shop" was done by David Balsam. Programming and graphics are by Martin Kahn, copyright 1984 by Broderbund Software. "Fantavision" was designed and produced by Scott Anderson. The Fantavision user's manual was written by Gary Carlston and Kay Wayland, copyright 1985 by Scott Anderson.

We also wish to thank Data Transforms, Inc. for permission to describe and demonstrate "Fontrix Version 1.5," the "Fontpak Character Set Disks," and "The Fontrix User's Manual," original copyright 1983, Data Transforms Inc., a division of Solarstatics Inc.

Thanks are also due to Beagle Brothers, Micro Software, Inc. for permission to reproduce programs and information contained in "Beagle Graphics, Double-Res Apple Graphics" by Mark Simonsen, Instruction Manual by Rob Renstrom and Bert Kersey, copyright 1984 by Mark Simonsen. Published by Beagle Brothers Inc.

The cooperation of these software publishers makes this new kind of book possible.

And finally, my thanks to Bob Prescott for his contribution of conversion programs, to Nick Francesco for technical assistance, to Bonnie Dennison for information regarding the Symtec video equipment, and to Lucy Prescott and Janet Schumacher for their editorial review of the book manuscript.

WILLIAM H. DEWITT

Section 1

The Programming Path to Low and HiRes Graphics

1

The Facts of Life with a Computer

This chapter deals with information you need to know before you start programming graphics. Although experienced programmers may regard this initial material as unnecessary, it is in fact the very material that I was seeking when I was trying to write my first graphics programs. If you're a beginner, this first chapter will help you avoid the pain, misery, and embarrassment of asking the first one thousand questions. (I've quit counting!)

Let's begin by taking a look at the two disk operating systems—DOS 3.3 and ProDOS—most frequently used to control the Apple II series computers.

A DOS 3.3 Overview

DOS 3.3 is the older of the two operating systems. Its major advantage over ProDOS is the reservoir of DOS 3.3 software available, compared to the fewer packages written for the newer system. However, this proportion will change, because ProDOS is clearly the system of choice for future software.

DOS 3.3 is easy to use. Any DOS 3.3 formatted disk can be cataloged to find out what's on it. However, no subdirectories (of the catalog) are possible with this system.

Another limitation of this system is that a disk formatted for DOS 3.3 can use only Apple Disk II drives. It cannot be used with a hard disk.

A ProDOS Overview

ProDOS functions at a much higher speed than does DOS 3.3.

An additional advantage of ProDOS is its ability to handle subdirectories. This feature enables the user to divide files into categories. In this connection, each ProDOS disk must be given a name (called a *volume name*) when formatted. (Refer to your Apple II manual, *Basic Programming with ProDOS*, for information on volume names and prefixes.)

ProDOS also achieves a certain amount of drive independence. It finds the file you want without having to be told in which drive the file resides.

The ProDOS system can be used with any Apple computer disk drive, including hard disk drives. The enormous memory capacity of a hard disk could be of great value for professional graphics use involving the storage of large numbers of pictures.

ProDOS formatted disks cannot be booted up unless they contain certain utility programs (which will be discussed later).

Summary

1. When you are working with software packages, the choice of operating system is made for you by the software publisher.
2. If you're doing your own programming, the choice of operating system is yours.
3. Neither of these two systems offers any functional graphics advantage.

4. Looking ahead a few years, you should expect that there will be a limit as to how far back any manufacturer's system support will reach, after new generations of hardware and software have been developed. (This expectation favors the more recently introduced ProDOS.)

Making a Choice

Since the operating systems used by various graphics software systems may be either DOS 3.3 or ProDOS as favored by the publishers, anyone contemplating graphics via software should be prepared to use either or both systems.

If you already have a quantity of programs and pictures stored in DOS 3.3 system diskettes, you may wish to maintain them in that system rather than do wholesale conversions to ProDOS. On the other hand, the subdirectory feature and faster operation of the ProDOS system make it the right choice if you are starting from scratch.

Now, let's move ahead to graphics programming information of a more general nature.

Get Organized

1. Assign a diskette to your graphics programs. You'll have a collection of them sooner than you think. Trying to find programs that are dispersed among several diskettes can be frustrating as well as time-consuming.

2. As you test the programs listed in this book, make it a point to SAVE (to diskette) each program *and* any variations you may try. This makes it possible to review the operation of a program without having to reenter it. You can always DELETE a program when you decide that you definitely don't want it anymore.

3. After you build up a bit of a library on your graphics diskette, COPY your most important programs onto a backup diskette. By doing this you will avoid any catastrophic loss should you have an accidental wipe-out.
4. If you have a printer, periodically print out the catalog of your graphics diskettes. Put the printouts in a notebook so that you can locate any program without having to catalog every diskette.

Be Economical

Within reason, it's a good idea to combine a number of programming instructions on each line. Not ten to a line, but if they're short, up to, say, four to a line. In a long program, the line numbers themselves use up quite a bit of memory. You can use colons to reduce the number of line entries, like this:

```
10 HOME : GR : COLOR= 9
20 X= 10 : Y= 15 : PLOT X,Y
30 END
```

Note: If you are writing a new program, it's best to avoid combining any terms that might be subject to adjustment.

Don't Be Intimidated

Most of the graphics commands for the Apple computer are mnemonically- or language-related to the action involved. However, you'll need to learn a few that are pure "computerese." Don't worry, there aren't too many, and there are reminders (REM statements) in the programs where commands like "CALL – 1998" or "POKE 49234,0" occur.

Suggestion: Make a list of these commands and keep it handy until you can remember the ones you use most frequently.

Pay Attention to Signs and Sequences

There are dozens of examples that can be cited to demonstrate the importance of sequence in graphics programming. In the following examples, don't worry about the exact meaning of the commands listed—they'll be fully explained later. The important factor is to develop an awareness that the *sequence* of commands can be critical.

1. The commands PLOT, HLIN, and VLIN will not function unless GR and COLOR= (a number) precede them.
2. In the high-resolution mode, similar sequential requirements apply: The commands HPLOT, SCALE, HCOLOR= (a number), and ROT must be preceded by HGR.
3. CALL – 1998 is a valid command. CALL 1998 without the minus sign will not work. Watch for those minus signs.
4. HGR followed by POKE 49234,0 clears the entire screen for high-resolution graphics. If POKE 49234,0 precedes HGR, the computer ignores it and leaves room for four lines of text at the bottom of the screen as it would for the HGR command alone.

A message from the computer: "Give me valid commands and remember that I'm fussy about their sequence."

Some Programming Tools

The programs of Chapter 2 use a number of simple but effective programming devices to draw a dotted line, to count, and to generate random numbers. Let's get aquainted with these routines before we use them to put images on the monitor. Try entering and running a couple of these mini-programs, just for batting practice.

A fundamental need in creating graphics images is the ability to change the value of a variable by specific increments over a specific range. This is easily done with the following program steps:

```
30 FOR X= 1 TO 50 : STEP 2
35 PRINT X :REM SO WE CAN WATCH X VALUES CHANGE
40 NEXT X
```

Note: You can reduce the value of a variable by using the term STEP – (a number) and starting at the high value of the variable:

```
10 FOR X = 30 TO 0 STEP - 2
20 NEXT X
```

Essentially the same device can be used for simple counting or timing purposes.

Random Numbers

Using random numbers to select the color to be plotted at a randomly selected point doesn't necessarily create art, but I've sure had fun doing it! You can get the Apple to generate random numbers for you by using the command RND with some other appropriate terms.

For graphics purposes it's best to use whole numbers or integers. The following expression is perhaps the simplest means of generating integers in the desired range.

```
X = INT ( RND (1) * THE NUMBER OF YOUR CHOICE )
```

For example:

```
X = INT ( RND (1) * 16 )
```

will generate a series of numbers from 0 to 15.

Note: To get whole numbers you must include the INT term. Remember also that the numbers generated are from 0 to *one less* than the multiplying number at the right of the expression.

A Notable Exception to Remember

Little is said in the Apple documentation or elsewhere about using the Page 2 display screen for low-resolution graphics. The reason is that BASIC will not function with Page 2. Beginners should not attempt to use this combination.

Now, let's go on to Chapter 2 and get some images on the screen.

2

Low-Resolution Graphics

The Low-Resolution Screen

The graphics display screen of your computer monitor is like an electronic billboard or grid. In the low-resolution graphics mode, this grid has 40 horizontal and 40 vertical points for a total of 1600 controllable points. Any point can be programmed "on" or "off" in any one of 16 colors. (Later we'll discuss how to increase the graphics screen area, but don't worry about that now.)

This chapter defines the primary low-resolution graphics commands and shows you how to use them.

GR Commands and Colors

Using only five low-resolution commands in simple BASIC programming, you can create colorful, fast-moving images on your monitor screen. Here they are:

- **GR** makes the computer "think" *graphics* instead of *text.*
- **PLOT** "turns on" a specified X,Y location. It can be used in a "FOR X= (a numerical sequence): NEXT X—" type of statement but it cannot be used in the form "PLOT X,Y TO X1,Y1"

```
Example: PLOT 20,20
Example: Y= 20 : FOR X= 0 to 10
         PLOT X,Y : NEXT X
```

- **HLIN** "draws" a line between two specified points on the X axis at a specified point on the Y axis.
  Example: HLIN 0,20 AT 25
- **VLIN** "draws" a line between two specified points on the Y axis at a specified point on the X axis.
  Example: VLIN 15,25 AT 22
- **COLOR** must be specified *before* the PLOT command. Colors are selected by number. Note that the command is COLOR= (a number).

Table 2.1 **Colors Available for Low-Resolution Graphics**

0 BLACK	4 DARK GREEN	8 BROWN	12 GREEN
1 MAGENTA	5 GRAY	9 ORANGE	13 YELLOW
2 DARK BLUE	6 MED. BLUE	10 GRAY	14 AQUA
3 PURPLE	7 LIGHT BLUE	11 PINK	15 WHITE

GR Commands at Work

Now let's move from definitions to some short "getting acquainted" programs that illustrate how these GR commands work.

Program 2.1 **Drawing Lines with HLIN, VLIN**

Using HLIN and VLIN gives you high-speed plotting for linear and rectilinear images. This program will put a neat yellow border around your screen.

```
1  REM PROGRAM 2.1 DRAWING LINES WITH HLIN, VLIN
10 HOME : GR
20 COLOR= 13
30 HLIN 0,39 AT 0
40 VLIN 0,39 AT 0
50 HLIN 0,39 AT 39
60 VLIN 0,39 AT 39
70 END
```

In Chapter 3 you'll see ways to shorten this program. You'll also see how to make an image appear to "vibrate" on your screen.

Text and Images

You can get text on the screen with your graphics images. Want to put a title on a graph? No problem! The GR command reserves room for four lines of text below the graphics area for this purpose. Here's how you do it:

Program 2.2 **Adding Text to Graphics**

Use a program similar to Program 2.2, and if you need a refresher on the use of HTAB and VTAB, see pages 154 and 161 in your *Applesoft* book.

```
1  REM PROGRAM 2.2 ADDING TEXT TO GRAPHICS
10 HOME : GR : COLOR= 13
20 FOR Y= 0 TO 39 STEP 10
30 HLIN 0,39 AT Y : NEXT Y
40 FOR X= 0 TO 39 STEP 10
50 VLIN 0,39 AT X : NEXT X
60 HTAB 1 : PRINT "0" ; : HTAB 10 : PRINT "10" ; : HTAB 20 :
   PRINT "20" ; : HTAB 30 : PRINT "30" ; : HTAB 39 : PRINT "40"
70 VTAB 23 : HTAB 15
80 PRINT "PLOT YOUR DATA"
```

Using the Entire GR Screen

It's easy to get more graphics screen area if you need it. Adding two more commands will do it. After GR in your program, add the command POKE 49234,0 followed by CALL −1998 as shown in this example:

```
10 HOME : GR
20 POKE 49234,0 :CALL −1998
30 REM ENTER THE REST OF YOUR PROGRAM.
```

This sequence of statements will give you a screen area defined by 40 horizontal points (0 to 39) and 48 vertical points (0 to 47). Now you can increase your vertical PLOT and VLIN ranges accordingly.

Note: POKE 49234,0 *makes the entire screen area available* for graphics, but to clear the text area completely to black, you must use CALL −1998 following the POKE command.

Where Do We Go from Here?

These two programs have purposely been short to keep the emphasis on what each command does. However, you can learn a lot by experimenting further with them.

Have fun seeing what happens when you change the values of X and Y, or when you change the color. You'll learn something with every change you make.

Now I hope you will LOAD, SAVE, and RUN the rest of the programs in this chapter, which are generally more sophisticated than those presented so far. Many of these programs were written for my film, *Musique Graphique*, in which dynamic images of the computer were rhythmically related to music (something I urge you to try). More on this in Chapter 6.

Low Can Be Plus or Minus

Even though low-resolution images are limited in detail, this mode of graphics has other attributes that make it attractive for business and technical purposes. The gamut of colors and the rate at which plotting occurs make this mode a fascinating one to use.

These same features make this mode an attractive one for the creation of dynamic abstract patterns. Many of this chapter's programs are of a dynamic nature in the sense that time becomes an important element of the viewer's subjective reaction to what is appearing on the screen.

In summary, if you want fast action and lots of color, low resolution is the way to go!

Color Calibration

Before you start using the rest of the programs in this chapter, be sure to take a look at the low-resolution colors. Program 2.3 will give you a good calibration on the gamut of colors available.

After you ENTER, SAVE, and RUN this program, adjust your monitor to match the colors shown on page 18 of the Applesoft Tutorial book.

Program 2.3 **Apple Color Bars**

```
1  REM PROGRAM 2.3 APPLE COLOR BARS
10  HOME : GR
20  FOR C= 0 TO 32 : COLOR= C / 2
30  HLIN 0, C AT C : VLIN 0, C AT C
40  NEXT C
50  HTAB 2 : PRINT "GR COLORS SHOWN WITH 0 AT THE"
60  HTAB 2 : PRINT "TOP LEFT, 15 AT THE BOTTOM."
```

Note: The values of C are used not only to obtain color numbers but also to determine the length of the horizontal and vertical color bars.

The beautiful collection of colors resulting from this program explains why I use random color displays in many of my programs. In my opinion, computer graphics without color can be pretty dull. So use color to add sparkle and excitement to your programs. Put color to work for you!

Plotting Dots

The following technique can be useful for diagrammatic as well as artistic purposes. Let's light up points all over the screen!

Using a random number routine for the choice of locations to be turned on and the color to be displayed gives punch to this program. As an added touch, we'll turn on 50 points one at a time, turn them all off at once, and then start over again. Sounds complicated? Not really, you can do it in about seven program lines, exclusive of REMARKS. (See Program 2.4.)

Program 2.4 **Random Dot Plots**

```
1  REM PROGRAM 2.4 RANDOM DOT PLOTS
10 HOME : GR
20 FOR N= 1 TO 50 : REM WE'RE GOING TO TURN ON 50
                       POINTS ON THE SCREEN.
30 COLOR= INT ( RND (1) * 16 )
40 REM RANDOM COLORS FROM 0 TO 15.
50 X = INT ( RND (1) * 40 ) : Y= INT ( RND (1) *
      40)
60 REM X AND Y WILL BE RANDOM POINTS FROM 0
   TO 39.
70 PLOT X,Y : REM PLOT EACH X,Y COMBINATION AS
                 CHOSEN
80 NEXT N : REM COUNT UP TO 50
90 GOTO 10 : REM PLAY IT AGAIN SAM!
```

Now you have learned how to create a little action on the screen by the use of the random PLOT, random COLOR selection, and the use of a loop from line 90 back to line 10 (which blanked the screen). Remember these techniques and incorporate them in your own programs.

With a modicum of creative thinking you can use what you have just learned to write new programs simulating shooting stars, snowstorms, or whatever—*Use your imagination!*

Want to make a "crazy quilt" pattern? Eliminate lines 20 and 80; change line 90 to read "GOTO 30".

Next Stop on the HLIN, VLIN

Are you ready for "Framed Internality"? Framed *what?* You'll see: it's HLIN and VLIN with *zap!*

Program 2.5 **Framed Internality**

```
1 REM PROGRAM 2.5 FRAMED INTERNALITY
10 HOME : GR
20 COLOR= INT ( RND (1) *16 )
30 FOR Y=8 TO 36 STEP 28
40 HLIN 10,28 AT Y
50 NEXT Y
60 FOR X=10 TO 28 STEP 18
70 VLIN 8,36 AT X
80 NEXT X
90 Y2= 8 + INT ( RND (1) * 28 )
100 HLIN 10,28 AT Y2
110 X2= 10 + INT ( RND (1) * 18 )
120 VLIN 8,36 AT X2
130 GOTO 20
```

Want to have some fun? RUN this program and play some fast music on your stereo. Ooops! That's in another chapter, way up ahead.

"Blocking In" Areas

If you want to "block in" or fill certain areas of the screen or the whole screen with a particular color, the commands HLIN and VLIN are your best bet in the low-resolution mode.

To "block in" an area, draw a series of horizontal lines at one vertical position after another. The HLIN, VLIN combination does this job quickly. Try Program 2.6; it will give you some ideas for your own programs.

Program 2.6 **Blocking in an Area**

```
1 REM PROGRAM 2.6 BLOCKING IN AN AREA
10 HOME : GR
20 COLOR= 6 : REM A MEDIUM BLUE
30 FOR Y= 0 TO 16 : HLIN 0,12 AT Y
40 NEXT Y
50 END
```

There you have it, a nice solid rectangle of color! Could this be the beginning of a flag? Part of a floor plan?

If you have an idea, draw it on graph paper to get the proportions approximately right and then write a program to make it happen on your monitor. Go ahead, you can do it!

Filling the Entire Screen with One Color

Tired of black backgrounds? Change line 30 of Program 2.6 and you can have any color you want (low-resolution color, that is).

Program 2.7 **Screen Color**

```
1 REM PROGRAM 2.7 SCREEN COLOR
10 HOME : GR
```

```
20  COLOR= 13 : REM (CHOOSE A COLOR NUMBER)
30  FOR Y= 0 TO 39 : HLIN 0,39 AT Y
35  REM LINE 30 ONLY CHANGE FROM PROGRAM 3.6
40  NEXT Y
50  REM CONTINUE WITH THE REST OF YOUR OWN
    PROGRAM!
```

Togetherness Really Works

Putting several commands to work together is what makes creative programming. Here's a program that illustrates how to make a dynamic flowchart by using a combination of low-resolution commands.

Using HLIN and VLIN you can make a box and a solid rectangle representing two steps of a process. Flow between the two steps is indicated by the flashing dotted lines.

You can use this "starter" information to make the kind of chart you need for a process or organization.

Program 2.8 **Flowchart**

```
1  REM PROGRAM 2.8 FLOWCHART
10  HOME : GR : COLOR= 2
20  FOR Y= 0 TO 16 STEP 16
30  HLIN 0,10 AT Y : NEXT Y
40  FOR X= 0 TO 10 STEP 10
50  VLIN 0,16 AT X : NEXT X
60  COLOR= 9 : FOR Y=0 TO 16
70  HLIN 25,35 AT Y : NEXT Y
80  FOR X= 11 TO 24 STEP 2
90  COLOR= 15 : PLOT X,2 : NEXT X
100  FOR X= 11 TO 24 STEP 2
110  COLOR= 0 : PLOT X,2
120  FOR N= 1 TO 50 : NEXT N
130  NEXT X : GOTO 80
```

Notice how lines 80 through 120 draw the dotted line. Drawing the line first in white, then in black, makes it appear to flash.

Using the "FOR N= 1 TO 150 : NEXT N" timing device *between* plots of X in black adjusts the drawing rate to give a sense of movement. This is a neat little trick to remember.

Curves Ahead

Whether your interest in curved lines is artistic or technical, you'll find that they have some unusual characteristics when plotted in the Apple computer's low-resolution mode. Stepwise jumps and odd overlaps (of two or more lines) that are technically undesirable may have a certain appeal from an artistic point of view.

The next program gives you a good idea of how the low-resolution mode handles curved lines.

Program 2.9 **A Curved Line**

```
1 REM PROGRAM 2.9 A CURVED LINE
10 HOME : GR
20 COLOR= 7
30 FOR X= 0 TO 39 STEP 2
40 Y= INT ( 39 - .02 * X * X )
50 PLOT X,Y : NEXT X
```

Not the smoothest curve you'll ever see, but it could convey lots of information.

For most technical applications the high-resolution mode is a better choice. However, for a strong, bold look, or for some artistic purposes, low resolution has its points (no pun intended).

More Low Resolution Later

Don't worry, that's not the end of the low-resolution programs. You'll find more in the Appendix at the back of the book.

Conceptual Programming

This is a good time to introduce conceptual programming. Basically, conceptual programming is using what you know to create what you want by telling the computer how, when, and where to do what.

In other words, you decide exactly what you want the computer to do, and then (step by step) you tell the computer how to do it.

Once you have created the basic design, you can adjust it here and there until it is exactly as you want it.

As you experiment with the rest of the programs in this book, think through the planning process you need to write each program. What do you want the end result to look like? How will you tell the computer to do it? You're well on your way to conceptual programming.

Onward and Upward to High-Resolution Graphics

Now, as the programs say, GOTO Chapter 3 for a new experience in the wonderful world of high-resolution graphics. I'm sure you will enjoy the challenging capabilities of the Apple in this versatile mode.

3

High-Resolution Graphics

A Whole New World

The resolution and drawing capabilities of the Apple II Plus computer in its high-resolution graphics mode offer practically endless possibilities for the creative artist or designer.

To put these capabilities to work you need to understand the high-resolution commands and how to use them in programs. This chapter acquaints you with some fundamental commands and introduces you to the fascinating field of high-resolution graphics.

Just Like Television

There are two display screens available in the high-resolution mode. They are referred to as Page 1 and Page 2.

Each high-resolution display screen is capable of displaying 54040 points of information. The screen can be compared to a television picture having a raster of 192 horizontal lines composed of 280 points per line. These points are called pixels. Every pixel can be turned on or off and programmed for any one of the six high-resolution colors.

High-Resolution Commands

To get you started in the high-resolution mode, we'll look at the commands in groups with appropriate working examples.

You have a choice of two commands to get your computer "thinking" high-resolution graphics.

- **HGR** sets Page 1 of high-resolution graphics with screen dimensions 280 by 160. It clears the screen to black, leaving room for four lines of text at the bottom. You can increase the graphics area to full screen (280 by 192) by following the HGR command with the POKE command, POKE – 16302,0 or its equivalent, POKE 49234,0.
- **HGR2** sets Page 2 of high-resolution graphics with a screen display 280 by 192. It clears the screen to black. Text at the bottom of Page 2 graphics is possible but not recommended for beginners.

Most demonstration programs use HGR (Page 1), which happens to be my preference. Further information on the choice and use of HGR/HGR2 is found in later chapters.

HGR Colors

As in the GR mode, colors are specified by number, as listed in Table 3.1.

Table 3.1 **High-Resolution Colors**

0	BLACK 1	4	BLACK 2
1	GREEN	5	ORANGE
2	VIOLET	6	BLUE
3	WHITE 1	7	WHITE 2

A Three-Way Command

HPLOT is a command that can be used in three different ways. Be sure to ENTER and RUN the sample programs that follow. They will reinforce your understanding and give you some good hands-on experience.

Important: HPLOT *must always* be preceded by HGR or HGR2 to avoid clobbering important memory areas.

To Plot a Dot

Here's the first of HPLOT's three definitions:

- **HPLOT X,Y** causes a dot to be plotted at the screen position represented by the values of X and Y. The color of the dot will be that of the last HCOLOR command.

Now you deserve the rich reward of putting this command to work. Here's a program to give your screen the look of a star-filled sky—well, almost.

Program 3.1 **Starry Sky**

```
1  REM PROGRAM 3.1 STARRY SKY
10  HGR : POKE 49234,0 : HCOLOR= 3
20  FOR N= 1 TO 150
25  REM THERE WILL BE 150 STARS (TONIGHT?)
30  X = INT (RND (1) * 279) : Y = INT (RND (1) * 190)
35  REM LINE 30 GENERATES RANDOM X,Y LOCATIONS
40  HPLOT X,Y
50  NEXT N
```

Don't be surprised if those "stars" in the "sky" are bluish or greenish—that's just the way your color TV reacts to the Apple II Plus plotting individual points.

The HPLOT Thickens

Drawing lines with the HPLOT command is easy—just connect pairs of X and Y values together in *series*, using the word TO, as shown here:

```
HPLOT X,Y TO X1,Y1 TO X2,Y2 TO X3,Y3—ETC.
```

The color of lines plotted in this manner will be that of the last HCOLOR command. To change to another color, just break the series, enter a new HCOLOR command, and start a new sequence.

ENTER and RUN this program for a neat violet border around your monitor screen.

Program 3.2 **Screen Border**

```
1  REM PROGRAM 3.2 SCREEN BORDER
10  HGR : POKE 49234,0 : HCOLOR= 2
20  X= 0 : Y= 0 : X1 = 270 : Y1 = 191
30  HPLOT X,Y TO X1,Y TO X1,Y1 TO X,Y1 TO X,Y
35  REM SEE FOLLOWING PARAGRAPHS
40  END
```

Another HPLOT Uncovered

The least-used form of HPLOT TO is HPLOT TO X,Y:

- **HPLOT TO X,Y** will plot a line from the last dot previously plotted to location X,Y. The color of the line will be that of the last dot plotted, even though an HCOLOR change has been made.

Program 3.5 later in this chapter demonstrates the usefulness of this command.

Try It Yourself

Now is a good time to get some further experience with the commands you have just learned, so let me suggest that you write a few more short programs to get used to the commands we've just defined. How about a program to draw a square or a rectangle?

Next, ENTER and RUN the demonstration programs that follow. After you RUN a program, make some of the changes suggested and try a few variations of your own. Remember, one of the fastest ways to learn graphics programming is to make changes in existing programs and see what happens!

There's nothing quite so fascinating as playing that game of "What would happen if . . .?" So be bold! Let your imagination run free—you'll find the results really exciting.

An important reminder: When you enter any program in your computer, be sure to save it before you run it. If you make it a practice to do this, you will avoid the possibility of having to reenter the whole program because you forgot to save it.

Demonstration Programs

For the most part, the demonstration programs that follow make use of only those graphics commands listed in this chapter in combination with simple BASIC commands like GOTO and FOR X= (A RANGE OF VALUES), NEXT X, and so forth. You'll find that any techniques used to create images are explained in the program REM statements or in program notes.

Don't worry about the trigonometric functions (SIN and COS) in some of the programs. They're like power steering and automatic transmission, in that you don't need to understand them to enjoy them.

The Art of Curves

This program shows you how to generate families of curves that can be used for either data presentation or artistic purposes. For data/graphics use, you will have to make modifications to fit your own data.

This program will also teach you some programming tricks worth noting for future reference. First, it shows how the screen location of curves can be altered according to your needs. Second, it demonstrates how forms (in this case, families of curves) can be combined to create attractive artistic designs.

Program 3.3 **The Art of Curves**

```
1  REM PROGRAM 3.3 THE ART OF CURVES
10  HGR : POKE 49234,0
15  REM DON'T LET "A" IN NEXT LINE WORRY YOU.
20  A= 0.0005 : N= 1
30  FOR X= 0 TO 279
40  Y= INT ( A * X * X )
50  IF Y > 170 THEN GOTO 90
60  HCOLOR= 6
65  REM NEXT 2 LINES LOCATE START OF CURVES.
70  HPLOT 279 - X, 170 - Y
80  HPLOT X, 170 - Y
90  NEXT X
100  A= A + 0.001
105  REM LINE 110 COUNTS SETS OF CURVES DRAWN
110  N= N + 1 : PRINT N
120  GOTO 30
```

If you let this program run until it has drawn 40 or 50 sets of curves, the result is an attractive conical pattern formed by the multiplicity of lines emerging from each corner.

Want a different point of view? Place both curve sets at the top of the screen by making these line changes:

```
70  HPLOT 279 - X,Y
80  HPLOT X,Y
```

Now that you know how to do it, place both curve sets at the same side of the screen with these line changes:

```
70 HPLOT X, 170 – Y
80 HPLOT X,Y
```

A suggestion: Write a program that will place one set of curves in each corner of the screen.

For your own experiments, the counting device, N= 1 and N= N + 1 of lines 20 and 110, is included. To see how many curves have been drawn, hit RESET and you will see the number at the left of your screen. (Remember that hitting RESET will destroy the picture.)

A Keep-It-Simple Circle

Want to draw a circle? Want to draw an ellipse? Polar coordinates got you down? Step right up to this program and by fooling around with *seven* program lines, you can draw all of the above and more.

The following program makes drawing a circle simple.

Program 3.4 **A Simple Circle**

```
1 REM PROGRAM 3.4 A SIMPLE CIRCLE
10 HGR : POKE 49234,0 : HCOLOR= 7
20 R= 80 : FOR I= 0 TO 6.28 STEP .09
30 X= R * SIN ( I )
40 Y= R * COS ( I )
50 HPLOT 140 + X, 80 + Y
60 NEXT I
```

You can change the number of dots around the circumference of the circle by varying the last number of line 20. For a fairly solid line, try a value of .01.

Want a bigger circle? Increase the value of R. Want a smaller circle? Right.

Changing the numerical values of line 50 will move your circle around the screen.

Now for you fast learners, let's draw an ellipse. Change line 40 to read:

```
40  X= R * SIN ( I ) / 2
```

See, you don't have to understand SINes and COSines to have a good time.

Some suggestions: Write a program that will generate a series of concentric circles. Write a program that will combine many circles to make a tubular form.

Now it's time to pick up a few other programming tricks.

A Calculated Look

The next program creates a striking image formed by a linear progression of circles.

When you RUN this program, you will see how reducing the amount of calculation the computer must do and the use of only one loop speeds up image formation or drawing on the screen.

Program 3.5 **Tubular Dynamics**

```
1  REM PROGRAM 3.5 TUBULAR DYNAMICS
10  HGR : POKE 49234,0 : HCOLOR= 7
20  R= 80
30  FOR I= 0 TO 6.28 STEP .03
40  X= R * SIN ( I )
50  Y= R * COS ( I )
60  HPLOT 80 + X, 92 + Y
70  HPLOT 90 + X, 90 + Y
80  HPLOT 100 + X, 88 + Y
90  HPLOT 110 + X, 86 + Y
100  HPLOT 120 + X, 86 + Y
```

```
110 HPLOT 130 + X, 88 + Y
120 HPLOT 140 + X, 90 + Y
130 HPLOT 150 + X, 92 + Y
140 HPLOT 160 + X, 94 + Y
150 HPLOT 170 + X, 96 + Y
160 HPLOT 180 + X, 98 + Y
170 HPLOT 190 + X, 100 + Y
180 HPLOT 200 + X, 102 + Y
190 NEXT I
```

Another approach to this program is to use FOR, NEXT loops to generate the arithmetic values to be added to X and/or Y, but its operation will be much slower.

Still another approach would be to enter these values as DATA and then READ the DATA in the right sequence. (This method is described in Chapter 4, which deals with shapes and shape tables.)

HPLOTting into Space

The last HPLOT program of this chapter is short but colorful and dynamic. There is a pulsar quality to the fast-changing image against a star-filled background.

Program 3.6 **Pulsar**

```
1 REM PROGRAM 3.6 PULSAR
10 HGR : POKE 49234,0
20 HCOLOR= INT ( RND ( 1 ) * 8 )
25 REM PUT STARS IN THE SKY
30 HPLOT INT ( RND ( 1 ) * 279 ), INT ( RND ( 1 ) * 190 )
40 X= INT ( RND ( 1 ) * 279 )
50 Y= INT ( RND ( 1 ) * 190 )
60 HPLOT X,Y
65 REM MAKE A PULSAR
70 HPLOT TO 140,90
80 GOTO 20
```

Some suggestions: Use some of the tricks you have already learned to move the PULSAR around the screen, to draw it in one place N times and then move it elsewhere, to make it all one color. When you try to change the size of it, you'll have to get your brain in gear.

For more HGR programs, see the Appendix at the back of the book.

Want to Double-Up?

If you have HGR pictures stored away, why not convert them to double HiRes? Program 12.4 of Chapter 12 will do the trick in seconds. Look to Chapter 13 for information on the conversion of HGR *programs* to double HiRes.

Now, on to Chapter 4 and greatly enhanced imaging possibilities with the use of shapes and shape tables.

4

Getting into Shape

Shape Defined

The ability of the Apple II Plus to deal with *shapes* is the key to its practically limitless artistic possibilities in the hands of the creative artist or designer. Many of the designs within this book make use of shapes.

What is a shape? What can you do with it? The next few paragraphs will deal with these questions in a nontechnical manner to help you get some idea of what a shape is before you try to use one in a program.

Think of a shape as a screen image that can be stored, retrieved, and drawn in any available color, rotated about an axis, and changed in size in accordance with program commands. Well, you're not really storing or retrieving *the image*—you're storing and retrieving a module of digitized information that represents an image.

By manipulation of one or more shapes in a program, you can create new forms or images. It is this exciting ability of the Apple computer that has never been described in a way to make its creative usefulness understandable by noncomputer science users.

Since the use of shapes is probably the most important feature of Apple graphics, this entire chapter is devoted to shapes and *shape tables.*

The purpose of this chapter on shapes is to give you a general understanding of the requirements for writing a shape program.

Learning to use the shape feature will enable you to create designs suitable for textiles, wrapping paper, technical purposes, or viewing as an art form.

Let's look at the commands used to DRAW and manipulate shapes.

Shape Commands

Four simple commands provide the key to the fantastic creative possibilities offered by the use of Shapes:

```
ROT= A NUMBER BETWEEN 0 AND 64
```

This command determines the rotational position of the shape. ROT= 0 will cause the shape to be drawn as it was defined. ROT= 16 will draw the shape 90 degrees clockwise. Each increment of 16 will rotate the shape another 90 degrees.

If SCALE= 1, only 4 rotational positions will be operational. If SCALE= 2, 8 rotational positions will be operational, and so forth. (ROT= can be a value of 8, 16, 24, etc.)

```
SCALE= A NUMBER FROM 1 TO 255
```

This command sets the scale size to be used for each DRAW and XDRAW command. (SCALE= 0 is *maximum* size. This is an unexplained anomaly of the Apple system.)

```
DRAW ( SHAPE # ) AT X, Y COORDINATES
```

This command will draw any shape at the given coordinates. The color, rotation, and scale for the shape must have been specified previously.

The shape number can be any number between 1 and 255. The X and Y coordinates can be any number from 0 to 278 and 0 to 191 respectively.

The final shape command, XDRAW, can be used alternately with DRAW to create some very unusual effects.

```
XDRAW ( SHAPE # ) AT X, Y COORDINATES
```

This command is essentially the same as DRAW, but the color of the shape will become what the *Applesoft Basic Programming Reference Manual* calls the *complement* of that originally plotted. The color changes are not actual complements, but the changes are spectacular anyway.

Another purpose of this command is to erase. If HCOLOR= 3 or HCOLOR= 7 (white), DRAWing a shape and then XDRAWing it will erase the shape without erasing the background.

Now let's get down to the nitty-gritty of making a shape and doing something with it.

About Shapes and Shape Tables

A shape is a module of digitized information representing an image. In this book all shapes are specified by decimal numbers in an array, or series of numbers, called a shape table. The position (first, last, etc.) of each number and its arithmetic value are determined by rules explained in Table 4.1.

Note: In this chapter we will deal exclusively with shape tables containing only one shape.

Protocol Forever

Here's a typical shape table. Its seven bytes of data will create one shape, a square.

```
LINE # X        DATA 1, 0, 4, 0, 44, 62, 0
```

All numbers up to the start of the first shape in any shape table are called the "header."

The second number in a shape table is always 0, because that position is never used.

Zeros are always used to indicate the end of one item of information. For example, the data representing the shape itself *always* ends with a 0.

The 0 tells the computer that it has just read the end of a shape.

Apple Bytes Shape Data

Now let's see how the computer understands what the numbers in the shape table mean. See Table 4.1.

Table 4.1

Byte #	**Data**	**Meaning to Computer**
1	1	Number of shapes in the table
2	0	Unused
3	4	Number of bytes to start of shape data
4	0	Unused
5	44	Data representing shape #1, a small square
6	62	
7	0	There is a 0 at end of every shape

Now that you have a shape table, you should put it in memory where the computer can find it (with a little programming help from you).

When an Apple computer encounters the word DRAW, it likes to look at memory locations decimal 232 and 233 for the *address* of any shape data. For that reason, we will put our shape data in a memory location and store the address of that location in 232 and 233. Here's how to do it.

We'll use the memory location decimal 7676 (LOC= 7676) as a starting address for our data. Then we'll tell the computer that our data starts at LOC= 7676 by appending the numbers 252 and 29 to 232 and 233 respectively. This is done in program lines as follows:

```
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7682
```

Note: The commas in line 1000 are necessary. LOC= is the abbreviation for LOCATION. Don't forget the = sign after LOC.

Add the number of bytes in the shape to 7676 to get the second LOC number. For example, 7676 plus 7 = 7682.

At this point, don't worry about why or how the POKE and LOC numbers are derived, just use them.

How the memory location decimal 7676 was chosen, why we use the number 252 and 29, and how shape data is generated is explained on pages 92–97 of the *Applesoft Basic Programming Reference Manual.*

In Shape at Last

Now you're ready to write your first shape program. After you have entered Program 4.1, SAVE it for future reference. See Figure 4.1 for a preview.

Note: Those commas in line 1030 are important.

Program 4.1 **A Square Shape**

```
1 REM PROGRAM 4.1 A SQUARE SHAPE
5 REM HANDLE THE SHAPE DATA FIRST
10 GOTO 1000
20 HGR : POKE 49234,0
30 HCOLOR= 3 : SCALE= 25
40 ROT= 0
45 REM DRAW SHAPE AT CENTER OF SCREEN
```

(continued on the next page)

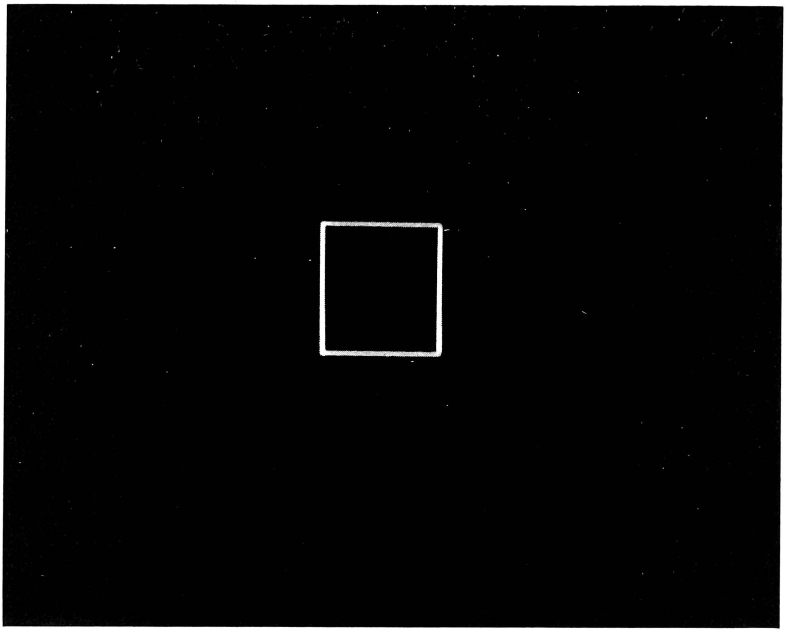


Figure 4.1 ***A Square Shape***

```
50  DRAW 1 AT 140, 80
60  RETURN
995  REM LOAD STARTING ADDRESS
1000  POKE 232,252 : POKE 233,29
1005  REM LIST LOC. TO BE USED
1010  FOR LOC= 7676 TO 7685
1015  REM READ THE DATA
1020  REM LOAD DATA INTO MEM. LOCS.
1025  READ BYTE : POKE LOC, BYTE
1030  DATA 1, 0, 4, 0, 56, 54, 45, 36, 7, 0
1035  REM PUT DATA IN ALL 7 LOC.
1040  NEXT LOC
1045  REM GET STARTED IN HIGH RES. MODE
1050  GOSUB 20
```

Ready for blast off? OK, RUN!

Voila! You now have a shape on your screen. An unimposing little square, but it can be made to create many other images as you will see in the shape Demonstration programs of Chapter 7.

Feeling creative? Write a program to DRAW this shape in a variety of sizes, in different colors, on diagonals across the screen, in changing rotational positions.

Now on to a program to convert our square shape into a ring, the data for several shapes, and two programs for fast evaluation of any shape data.

Shape Table Details

Learning how to make new and exciting images from a shape is fun. It's not only fun, it's an experience that will give you a million ideas to try.

The following pages will show you how to make a ring out of a square, how to evaluate shape data quickly, and how to put more than one shape in a shape table. You'll soon be off and running with shapes.

Rounding the Square

You can convert the square shape of Program 4.1 into a ring with no trouble at all. While the square shape itself could be the source of an interesting abstract design to fill your monitor screen, making a ring out of it offers other appealing design possibilities.

Instead of just making a ring out of a square, why not put a little frosting on the cake and draw rings in all available colors at random locations all over the screen? Try it! Suddenly you're a textile designer, an artist!

Figure 4.2 should make you a believer.

Program 4.2 **Rounding the Square**

```
1 REM PROGRAM 4.2 ROUNDING THE SQUARE
10 GOTO 1000
20 HGR : POKE 49234,0
30 HCOLOR= INT ( RND (1) * 8 )
40 X= INT ( RND (1) * 279 )
50 Y= INT ( RND (1) * 190 )
60 REM OTHER SCALES MAY BE AMAZING
70 SCALE= 20
80 REM ROTATING THE SQUARE MAKES A RING
90 FOR R= 0 TO 63 : ROT= R
100 DRAW 1 AT X,Y : NEXT R
110 REM MAKE MORE RINGS AT NEW COORDINATES
120 IF R= 63 THEN GOTO 30
130 REM OTHER HALF OF GOSUB
140 RETURN
995 REM TELL THE COMPUTER WHERE TO LOOK FOR DATA
1000 POKE 232,252 : POKE 233, 29
1010 REM THERE ARE 10 BYTES OF DATA TO BE STORED
1020 FOR LOC= 7676 TO 7685
1300 READ BYTE : POKE LOC, BYTE
1040 DATA 1, 0, 4, 0, 56, 54, 45, 36, 7, 0
1050 NEXT LOC
1060 GOSUB 20
```

The random nature of this program makes for some rather striking effects. If you want to photograph the screen, use CTRL C to interrupt the program. Enter CONT to continue DRAWing rings.

Suggestion: As an interesting exercise, try some other shapes in this program. An attractive wreathlike ring is created by using the “loose square” shape listed in Table 4.2. (See page 45.)

Be sure to make the appropriate adjustments in LOC and DATA values.

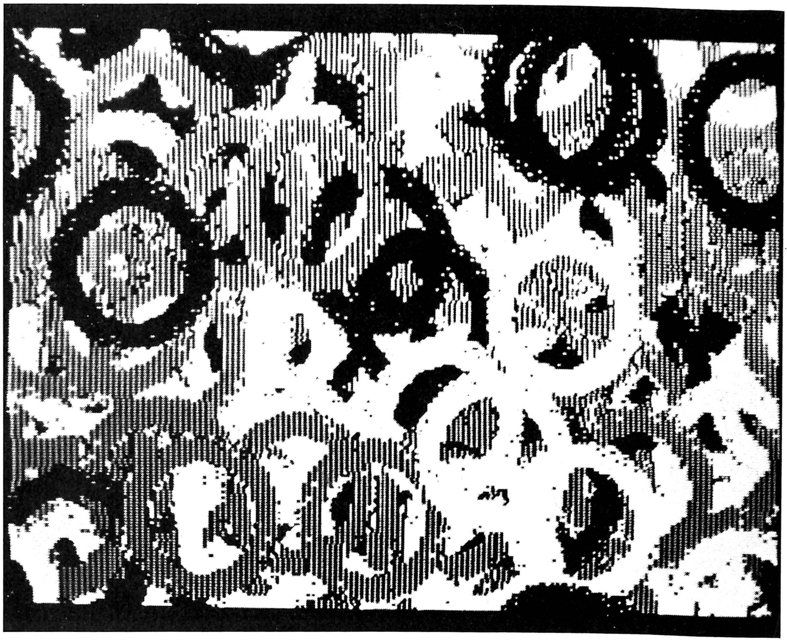


Figure 4.2 ***Rounding the Square***

Shape Up or Ship Out?

Want to see more shapes—fast? Need a program for easy shape evaluation? Relax, you're in exactly the right place!

You can enter the data from Table 4.2 into Program 4.3 to put shapes on the screen as fast as you can enter the numbers.

Notice that this program will handle and display more than one shape at a time. If your DATA includes too many shapes, you will get an error message indicating an "illegal HPLOT" because you have run out of room on the screen.

ENTER and SAVE the program. You can use it for a fast look at any shape that comes along. (See Figure 4.3.)

Program 4.3 **The Shape Tester**

```
1 REM PROGRAM 4.3 THE SHAPE TESTER
5 SPEED= 150
10 PRINT "HOW MANY NUMBERS"
20 PRINT "IN YOUR SHAPE DATA?"
25 PRINT " "
30 PRINT "ENTER THIS NUMBER"
35 PRINT " "
40 INPUT I : I= I - 1
45 PRINT " "
50 PRINT "HOW MANY SHAPES ?"
55 PRINT " "
60 INPUT J
70 GOTO 1000
80 HGR : POKE 49234,0
90 HCOLOR= 3 :SCALE=25 : ROT= 0
100 Z= 60
110 FOR N= 1 TO J
120 DRAW N AT Z,80 : Z= Z + 80
130 NEXT N
140 RETURN
1000 POKE 232,252 : POKE 233,29
1005 PRINT " "
1010 FLASH
1020 PRINT "ENTER YOUR SHAPE DATA"
1030 FOR LOC= 7676 TO 7676 + I
1040 INPUT X
1050 POKE LOC, X
1060 NEXT LOC
1070 GOSUB 80
```

Carousel, Anyone?

For those instances when you would like to see what kind of an image a shape makes when it is rotated, Program 4.4 will do the trick. This program displays the shape without rotation, with

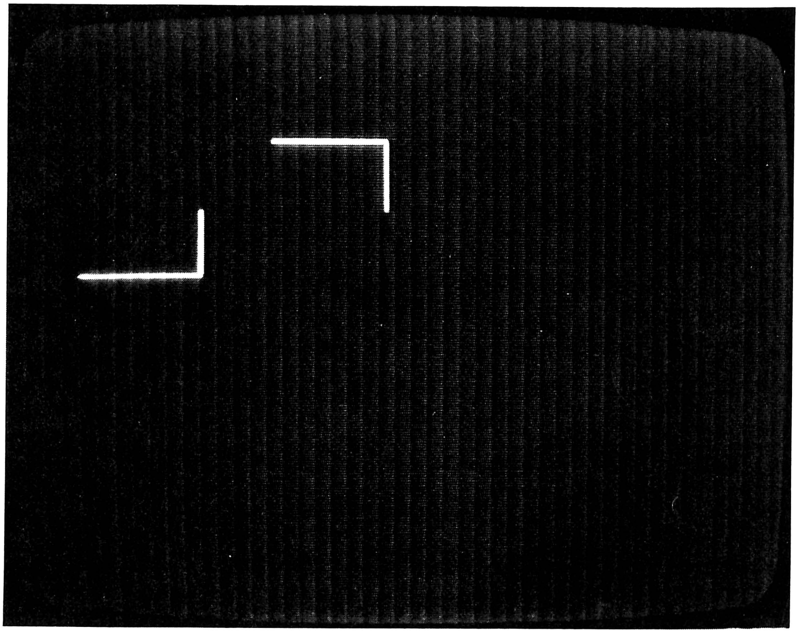


Figure 4.3 ***Shape Test***

continuous rotation, and with rotation at a STEP 4 increment. See Figure 4.4.

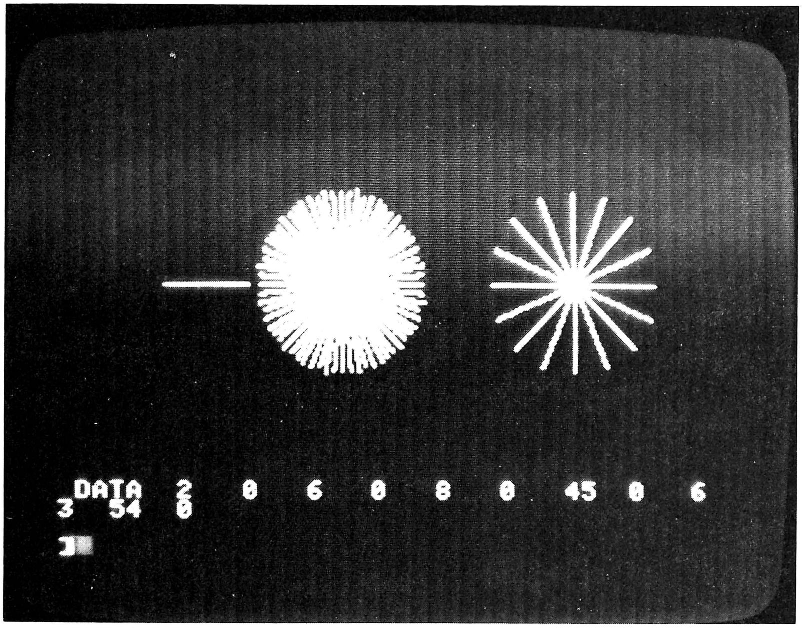
If you enter and SAVE this program for future reference, you will avoid the necessity of writing a test program to check out each new shape idea.

Program 4.4 **Rotational Shape Test**

```
1  REM PROGRAM 4.4 ROTATIONAL SHAPE TEST
10  HOME : SPEED 175 : K= 8
20  PRINT "THIS PROGRAM SHOWS YOU HOW A SHAPE
    LOOKS WITH AND WITHOUT ROTATION"
25  REM PRESS SPACE BAR BETWEEN QUOTES
30  PRINT " "
40  PRINT "PLEASE CHOOSE A SCALE. 10 IS GENERALLY OK"
50  INPUT S : SCALE= S
60  PRINT " "
70  PRINT "HOW MANY NUMBERS"
```

(continued on the next page)

```
80  PRINT "IN YOUR SHAPE DATA?"
90  PRINT "PLEASE ENTER THIS NUMBER"
100  INPUT I : I= I - 1
110  PRINT " "
120  GOTO 1000
130  HGR : HCOLOR= 3
140  ROT= 0 : DRAW 1 AT 45, 80
150  FOR R= 0 TO 63 : ROT= R
160  DRAW 1 AT 120, 80 : NEXT R
170  FOR T= 0 TO 63 : ROT= T
180  DRAW 1 AT 220, 80 : NEXT T
190  SPEED = 255 : END
1000  POKE 232,252 : POKE 233, 29
1010  PRINT "NOW ENTER YOUR SHAPE DATA"
1020  FOR LOC= 7676 TO 7676 + I
1030  VTAB 20 : INPUT X
1040  VTAB 21 : HTAB 2 : PRINT "DATA"
1050  VTAB 21 : HTAB K : PRINT X
1060  POKE LOC, X : K= K + 4
1070  NEXT LOC
1080  GOTO 130
```



```
 DATA  2   0   6   0   8   0  45   0   6
3  54  0
]
```

Figure 4.4 *Shape Rotation Test*

You'll be using this nifty way to evaluate shapes again and again, so be sure to SAVE it.

Notice the conversational touches added to Program 4.3 in writing Program 4.4. It does require extra work. However, this exercise may give you some ideas on how to add a little polish to your own interactive programs.

Now all you need is some data to plug into the program.

Shape Tables Galore

Table 4.2 shows something you seldom see, a tableful of shape tables. (No wonder people are intimidated by computers.)

Table 4.2 **Shape Data for Five Shapes**

Shape	# of Locations Required	Data
Large Square	10	1,0,4,0,56,54,45,36,7,0
Small Square	7	1,0,4,0,44,62,0
Straight Line	7	1,0,4,0,15,5,0
Loose Square	14	1,0,4,0,18,63,32,100,45,21,54,30,7,0
Cross	11	1,0,4,0,44,46,62,62,60,44,0

Use Table 4.2 to get the data requested for Programs 4.3 and 4.4. Enter it as directed and you'll have a shape on the screen in a flash.

Getting Shapes Together

Having covered the protocol or ground rules for a shape table containing only one shape, it's time to see what it takes to put two or more shapes into a shape table.

You can see from Table 4.2 that the headers for all of the shape tables containing one shape are the same: 1, 0, 4, 0. These headers change according to specific rules as the number of shapes in a table change.

If we put the large and small squares of Table 4.2 into one shape table, the data sequence looks like Table 4.3.

Table 4.3

Byte #	Data	Description
1	2	Number of shapes in table
2	0	Unused
3	6	Index to start of shape 1
4	0	Part of index
5	9	Index to start of shape 2
6	0	Part of index
7	44	Start of shape 1
8	62	Part of shape 1
9	0	End of shape 1
10	56	Start of shape 2
11	54	Part of shape 2
12	45	Part of shape 2
13	36	Part of shape 2
14	7	Part of shape 2
15	0	End of shape 2

Now you can see that the answer to the first question of Program 4.3 is 15 (bytes to be stored). Follow that up with the rest of your data and you'll see your two shapes like two old friends side by side on the screen. Isn't technology wonderful?

Getting the Picture

Shape tables do have a lot of numbers, but you can use Table 4.2 as a data source and write simple programs using one shape. Once you get the hang of it, you can start working with two or more shapes.

Demonstrating Shapes

Clearly there are many occasions when having a multiplicity of shapes available is important. However, you should recognize that much can be done using a single shape.

This point is also amply illustrated in the following demonstration programs, which feature programs that I have developed over the last few years.

Now, onward and upward. If you like abstract designs, you're going to love these programs.

Triangular Geometrics

Whether your designs are structured or random, the Apple's "shape" feature will help bring them to your monitor screen. The programs that follow illustrate the great variety of effects that can be obtained by manipulation of shapes.

The next shape demonstration program features a multiplicity of triangles, with variations in orientation that produce other geometric forms of an unusual nature. It's great fun to watch these forms evolve.

When you RUN this program, you will see some unusual colors on your screen. The reason for these occurrences is that writing one color over another sometimes creates colors not normal to the high-resolution mode.

Program 4.5 **Triangular Geometrics**

```
1  REM PROGRAM 4.5 TRIANGULAR GEOMETRICS
10  GOTO 1000
20  HGR : POKE 49234,0
30  A= INT ( RND (1) * 8 )
40  IF A= 1 OR A= 4 THEN A= 5
50  HCOLOR= A
60  X= INT ( RND(1) * 280 )
70  Y= INT ( RND(1) * 191 )
80  ROT= Z
90  FOR Q= 1 TO 40 : SCALE= Q
100  DRAW 1 AT X,Y
110 NEXT Q
120 Z= Z + 16 : IF Z > 64 THEN Z= 0
130 GOTO 30
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7682
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1,0,4,0,32,45,0
1040 NEXT LOC
1050 GOTO 20
```

If you are familiar with the works of the abstract artists Piet Mondrian and Ad Reinhardt, two possible shape data changes in Program 4.5 should pique your interest:

1. Substituting the straight-line shape data from Table 4.2 in line 1030 will result in images reminiscent of Mondrian's famous "plus-and-minus" paintings.
2. Using the shape data for the small square will produce an ever-changing screenful of rectilinear patterns which may remind you of Reinhardt's work.

Corridorically Speaking

Here are two programs that demonstrate how you can create the illusion of perspective using only the HPLOT and DRAW commands.

Program 4.6 **Doorless Corridor**

```
1 REM PROGRAM 4.6 DOORLESS CORRIDOR
10 GOTO 1000
20 HGR : POKE 49234,0
30 HCOLOR= 2 : ROT= 0 : Y= 25
40 FOR S= 1 TO 90 : SCALE= S
50 DRAW 1 AT 140, 100
60 NEXT S : HCOLOR= 0
70 FOR X= 60 TO 220 STEP 3
80 HPLOT X, 190 TO 140, 100
90 NEXT X
100 HCOLOR= 6
110 FOR T= 1 TO 90 STEP 12 : SCALE= T
120 DRAW 1 AT 140, 100 : NEXT T
130 RETURN
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7685
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1,0,4,0,56,54,45,36,7,0
1040 NEXT LOC
1050 GOSUB 20
```

Program 4.7 starts with a sequence of two engaging rectangular designs which then evolve into another corridor-like pattern.

For a longer look at the rectangles at the beginning of the program, add these time delays:

```
55 FOR N= 1 TO 1000 : NEXT N
75 FOR N= 1 TO 1000 : NEXT N
```

Program 4.7 **Another Corridor**

```
1 REM PROGRAM 4.7 ANOTHER CORRIDOR
10 GOTO 1000
20 HGR : POKE 49234,0
30 HCOLOR= 5 : ROT= 0 : Y= 25
40 FOR S= 1 TO 90 STEP 6 : SCALE= S
50 DRAW 1 AT 140, 100 : NEXT S
```

(continued on the next page)

```
60  FOR T= 90 TO 1 STEP -1 : SCALE= T
70  XDRAW 1 AT 140, 100 : NEXT T
80  HCOLOR= 0
90  FOR X= 60 TO 220 STEP 3
100  HPLOT X, 190 TO 140, 100
110  NEXT X
120  HCOLOR= 5
130  FOR U= 1 TO 90 STEP 12 : SCALE= U
140  XDRAW 1 AT 140, 100 : NEXT U
150  RETURN
1000  POKE 232,252 : POKE 233,29
1010  FOR LOC= 7676 TO 7685
1020  READ BYTE : POKE LOC, BYTE
1030  DATA 1,0,4,0,56,54,45,36,7,0
1040  NEXT LOC
1050  GOSUB 20
```

Spacewalking Shapes

The last few of these shape demonstration programs make extensive use of the XDRAW command to produce illusionary effects of lines and planes in space.

You probably don't need any urging to try DRAW instead of XDRAW, so go ahead—try a few. You'll see a very different kind of image. Use DRAW and XDRAW alternately for more fun.

Now let's see how the Apple does in outer space.

Program 4.8 **Magic Wands**

```
1  REM PROGRAM 4.8 MAGIC WANDS
10  GOTO 1000
20  HGR : POKE 49234,0
30  HCOLOR= 3 : SCALE= 10 : ROT= 8
40  FOR X= 40 TO 200 : Y= 145
50  XDRAW 1 AT X,Y : NEXT X
```

```
60 FOR X= 180 TO 0 STEP -1 : Y= 145
70 XDRAW 1 AT X,Y : NEXT X
80 GOTO 40
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7690
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1,0,4,0,63,63,63,63,63,63,63,63,63,63,0
1040 NEXT LOC
1050 GOTO 20
```

Now for a translation from lines to planes—or are they?

Program 4.9 **Rotating Plane**

```
1 REM PROGRAM 4.9 ROTATING PLANE
10 GOTO 1000
20 HGR : POKE 49234,0
30 HCOLOR= 3 : ROT= 8 : SCALE= 10
40 FOR X= 0 TO 140 : Y= 140
50 XDRAW 1 AT X,Y : NEXT X
60 FOR X= 140 TO 1 STEP -1
70 Y= 140
80 XDRAW 1 AT X,Y
90 NEXT X : GOTO 30
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7690
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1,0,4,0,32,32,32,32,32,32,32,32,32,32,0
1040 NEXT LOC
1050 GOTO 20
```

There's that old question, "If one is good, would two be better?" For a quick answer, RUN the next program.

Program 4.10 **Dual Rotating Planes**

```
1 REM PROGRAM 4.10 DUAL ROTATING PLANES
10 GOTO 1000
20 HGR : POKE 49234,0
30 HCOLOR= 3 : ROT= 8 : SCALE= 7
40 FOR X= 50 TO 100 : Y= 120
50 XDRAW 1 AT X,Y : NEXT X
60 FOR X= 160 TO 80 STEP -1
70 Y= 120
80 XDRAW 1 AT X,Y
90 NEXT X : GOTO 30
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7690
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1,0,4,0,32,32,32,32,32,32,32,32,32,32,0
1040 NEXT LOC
1050 GOTO 20
```

You can speed up the operation of this program considerably by reducing the number and size of the images on the screen.

Try reducing the scale size. Change lines 1010 and 1030 to read:

```
1010 LOC= 7676 TO 7681
1030 DATA 1,0,4,0,32,0
```

Program 4.11 is intended to stimulate your interest in dynamic programming and to illustrate how action on the screen holds your attention. Build action into your programs. Keep those shapes moving, not only horizontally but vertically as well.

Program 4.11 **Crosses Crossing**

```
1 REM PROGRAM 4.11 CROSSES CROSSING
10 GOTO 1000
20 HGR : POKE 49234,0 : HCOLOR= 3
30 HPLOT 0.0 : CALL 62454
40 ROT= 8 : SCALE= 20
50 FOR X= 60 TO 180 : Y= 180
60 XDRAW 1 AT X,Y
70 NEXT X
80 FOR X= 60 TO 180 STEP 2
90 Y= 80 : HCOLOR= 6
100 DRAW 1 AT X,Y
110 NEXT X
120 FOR X= 180 TO 60 STEP - 2
130 Y= 80
140 XDRAW 1 AT X,Y : NEXT X
150 FOR Y= 20 TO 140 : X= 120
160 HCOLOR= 5
170 XDRAW 1 AT X,Y : NEXT Y
180 GOTO 40
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7686
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1,0,4,0,44,46,62,62,60,44,0
1040 NEXT LOC
1050 GOTO 20
```

5

Animation

Some Common Ground

Using animation techniques, you can add humor and excitement to your programmed drawings. You'll find that the techniques learned in this chapter will help you build action into new programs as you write them.

Animation can be described as the sequential display of a series of similar images at a rate fast enough to create the illusion of movement or other change.

Apple Picking

The fastest way to display and manipulate images is by the use of machine language. However, since this book deals only with BASIC programming, we'll stay with that language and show you how to use the Apple's ability to handle shapes for animation purposes.

Long May She Wave

Here's an American flag that will ripple in the breeze. Program 5.1 tells the computer to draw seven red and six white stripes of appropriate dimensions, a blue field of certain proportions, and 50 white stars on the blue field. Let's agree that the stars can be somewhat symbolic.

Lines 270 to 400 alternately widen and narrow the stripe edges and the bottom of the blue field to create a slight rippling effect.

Program 5.1 **Old Compuglory Ripples**

```
1  REM PROGRAM 5.1 OLD COMPUGLORY RIPPLES
10  A= 0 : B= 13 : C= 14 : D= 26
20  GOTO 1000
30  HGR : POKE 49234,0
35  REM DRAW RED STRIPES
40  HCOLOR= 5 : ROT= 0 : SCALE= 140
50  X= 140 : FOR Y= A TO B
60  DRAW 1 AT X, Y
70  IF Y > 167 THEN GOTO 110
80  NEXT Y
90  A= A + 26 : B= B + 26
100  GOTO 50
105  REM DRAW WHITE STRIPES
110  HCOLOR= 3 : X= 140
120  FOR Y= C TO D
130  DRAW 1 AT X, Y
140  IF Y > 154 THEN GOTO 180
150  NEXT Y
160  C= C + 26 : D= D + 26
170  GOTO 120
175  REM DRAW BLUE FIELD
180  HCOLOR= 2 : SCALE= 65
190  X = 65 : FOR Y= 0 TO 91
200  DRAW 1 AT X, Y : NEXT Y
205  REM DRAW WHITE STARS
```

```
210 HCOLOR= 3 : SCALE= 5
220 FOR X= 10 TO 120 STEP 12
230 FOR Y= 12 TO 76 STEP 16
240 FOR R= 0 TO 63 STEP 8 : ROT= R
250 DRAW 1 AT X, Y : NEXT R
255 REM 260 COMPLETES FLAG
260 NEXT Y : NEXT X
265 REM SHORT STRIPE RIPPLE
270 HCOLOR= 3 : SCALE= 75 : ROT= 0
280 X= 204 : FOR Y= 14 TO 79 STEP 13
290 DRAW 1 AT X, Y : NEXT Y
295 REM LONG STRIPE RIPPLE
300 HCOLOR= 5 : FOR Y= 14 TO 79 STEP 13
310 DRAW 1 AT X,Y : NEXT Y
320 HCOLOR= 5 : SCALE= 140
330 X= 140 : FOR Y= 117 TO 168 STEP 13
340 DRAW 1 AT X, Y : NEXT Y
350 HCOLOR= 3 : FOR Y= 117 TO 168 STEP 13
360 DRAW 1 AT X, Y : NEXT Y
365 REM BLUE FIELD RIPPLE
370 X= 65 : Y= 91 : SCALE= 65
380 DRAW 1 AT X, Y
390 HCOLOR= 2 : DRAW 1 AT X, Y
400 GOTO 270
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7682
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1, 0, 4, 0, 15, 5, 0
1040 NEXT LOC
1050 GOTO 30
```

Now that you have entered all that code, don't forget to SAVE this program with its little animation trick for future reference.

You can use the techniques included here to make eyes blink, create facial movements in cartoon images, or for other animation purposes. Keep these techniques in mind for use with other shapes, whether created from data or entered from a graphics tablet.

Casso Comes to Life

You'll have loads of fun when you meet the cartoon character waiting to greet you in Program 5.2. You can make a little fellow named Casso come to life on your monitor screen with a minimum amount of programming effort.

First, why not give Casso's spirits a lift by giving him a neck to hold his head up? The familiar "fill-in" procedure shown in lines 25 through 40 will do that in a hurry.

Any computer person worthy of the name deserves a space-age home. The simple HPLOT TO and FOR/NEXT techniques in lines 45 through 100 will quickly build a castle for Casso.

Let's get conceptual. How do you draw a head? You know that a head isn't round. What you want is sort of an egg shape. Why not draw two white disks so that they overlap and form that egg shape? See lines 105 through 140.

We'll use a little artistic license and make Casso's eyes round (disks) and space them to make them look cartoon-reasonable. Imagine having HCOLOR= 2 eyes! However you describe them, blue eyes will be most becoming on our little friend. See lines 145 through 180.

A simple solution to a nose for Casso could be a straight line on an angle. Well, why not? You get a whole nose for only four program lines, 185 through 210.

Still using the same shape, we can make a thatch of hair by some further judicious rotating. You can choose any color except black for the hair. (Black blends into the screen.) The secret to adding hair to a bald head is in lines 215 through 260.

Now about the mouth. A well-chosen SCALE and some eyeballing for the location can provide a nifty mouth for our little man. Needless to say, we'll use HCOLOR= 5 for the mouth.

As indicated in the REM statements, variations in the color and rotation of shape #1 simulate speech movements for Casso's mouth. By providing one unchanging mouth line, you can avoid having an empty space in the mouth location as color and rotational changes occur. See lines 265 through 370.

DRAWing and XDRAWing shape #1 as alternate black and white spots in Casso's eyes add a further humanizing touch to this cartoon character. See lines 375 through 420.

You can adjust Casso's mouth and eye actions to fit the speech pattern of your choice with the timing loops included in LINES 390, 400, 410, 420, 430, and 440 (a point to consider if you're going to videotape a cartoon character and "dub in" sound).

You're going to have a lot of fun making Casso "talk." Further, you'll get ideas of value for other animation programs of your own. Remember, each action can be broken down into simple steps like those used here. Now, enjoy RUNning Program 5.2.

Program 5.2 **The Incomparable Casso Talks**

```
1 REM PROGRAM 5.2 THE INCOMPARABLE CASSO TALKS
10 HOME : GOTO 1000
20 HGR : HCOLOR= 3
25 REM DRAW NECK
30 FOR Y= 110 TO 135 : HPLOT 128, Y TO 152, Y
40 NEXT Y : HCOLOR= 5
45 REM CASSO'S SPACEAGE HOME
50 HPLOT 110,40 TO 180,45
60 FOR X= 157 TO 190 : HPLOT X, 135 TO 180,45
70 NEXT X : HPLOT 190,135 TO 40,135
80 FOR X= 40 TO 70 : HPLOT X, 135 TO 110,40
90 NEXT X : HCOLOR= 6 : FOR Y= 110 TO 135
100 HPLOT 72,135 TO 163, Y : NEXT Y
105 REM FORM HEAD SHAPE
110 X= 140 : FOR Y= 80 TO 90 STEP 10
120 HCOLOR= 3 : SCALE= 25
130 FOR R= 0 TO 64 : ROT= R
140 DRAW 1 AT X, Y : NEXT R : NEXT Y
145 REM BLUE EYES NEXT
150 HCOLOR= 2 : SCALE= 4
160 FOR X= 130 TO 144 STEP 14
170 Y= 75 : FOR R= 0 TO 64 : ROT= R
180 DRAW 1 AT X, Y: NEXT R : NEXT X
```

(continued on the next page)

```
185  REM NOSE NEXT
190  X= 134 : Y= 84
200  HCOLOR= 5 : SCALE= 10 : ROT= 2
210  DRAW 1 AT X, Y
215  REM HAIR NEXT
220  X= 145 : FOR Y= 60 TO 63 STEP 3
230  FOR R= 0 TO 9 : ROT= R : HCOLOR= 1
240  FOR S= 1 TO 20 : SCALE= S
250  DRAW 1 AT X, Y : NEXT S : NEXT R
260  NEXT Y
265  REM START OF MOUTH
270  X= 135 : Y= 96 : HCOLOR= 5
280  SCALE= 10
290  ROT= INT ( RND ( 1 ) * 5 )
300  DRAW 1 AT X, Y
310  H=INT ( RND ( 1 ) * 8 ) : IF H= 0 OR H= 4 THEN H= 3
320  IF H= 1 THEN H= 7
330  IF H= 2 THEN H= 5
340  HCOLOR= H
350  DRAW 1 AT X, Y
355  REM FIXED MOUTH LINE
360  HCOLOR= 5 : ROT= 3
370  DRAW 1 AT X, Y
375  REM EYE MOVEMENT
380  ROT= 0 : HCOLOR= 0 : SCALE= 2
390  DRAW 1 AT 130, 75 : FOR N= 1 TO 40
400  NEXT N : XDRAW 1 AT 130, 75
410  DRAW 1 AT 144, 75 : FOR N= 1 TO 30
420  NEXT N : XDRAW 1 AT 144, 75
425  REM VARY TALK SPEED
430  FOR N= 1 TO INT ( RND ( 1 ) * 100 )
440  NEXT N
450  GOTO 270
1000  POKE 232,252 : POKE 233,29
1010  FOR LOC= 7676 TO 7682
1020  READ BYTE : POKE LOC, BYTE
1030  DATA 1,0,4,0,15,5,0
1040  NEXT LOC
1050  GOTO 20
```

Horticultural Happenings

Now let's use conceptual programming principles to portray butterflies flying about a colorful bed of flowers on your monitor screen. In a relatively short program, you're going to use already-learned features to create this illusion.

A key element in this program is the use of two shapes. The data for these shapes (in line 1030) is from Table 4.2 (page 45).

Plantin' Time

How do you "plant" a colorful garden on your screen? No problem!

Back in Chapter 2, you learned how to generate random locations and colors. Also, from Program 4.4, Rotational Shape Test, you know how simple it is to DRAW a flower-like image using any one of several shapes. In this case, shape #1 will be your flower-maker.

You know how to make a flower, how to make lots of flowers of different colors, and how to put them all over the screen. What you need now is a flutter of butterflies.

Butterflies? Shape #2 draws an equilateral check mark. You can use this shape to make butterflies.

If you repeatedly DRAW and XDRAW shape #2 (in white) at a variety of orientations at random locations, butterflies will fly from flower to flower.

Two simple tricks enhance the illusory quality of this program:

1. In line 160, DRAWing and XDRAWing SHAPE #2 twice in locations just one X and one Y value apart makes the butterfly image slightly "heavier" for better visibility.
2. The FOR/NEXT counter of line 170 makes the butterflies appear to pause briefly as they move about the flowers.

Program 5.3 **Butterflies and Blossoms**

```
1 REM PROGRAM 5.3 BUTTERFLIES AND BLOSSOMS
10 D= 1 : GOTO 1000
20 HGR : POKE 49234,0 : SCALE= 10
30 A= INT ( RND ( 1 ) * 8 )
40 IF A < 1 OR A= 4 THEN A= 5
50 HCOLOR= A
60 X= INT ( RND ( 1 ) * 279 )
70 Y= INT ( RND ( 1 ) * 190 )
80 FOR R= 0 TO 63 STEP 4 : ROT= R
90 DRAW 1 AT X,Y : NEXT R
95 REM LIMIT THE NUMBER OF FLOWERS TO 80
100 D= D + 1 : IF D > 80 THEN GOTO 120
110 GOTO 30
115 REM START OF BUTTERFLIES
120 SCALE= 7 : HCOLOR= 3
130 X= INT ( RND ( 1 ) * 279 )
140 Y= INT ( RND ( 1 ) * 190 )
150 ROT= INT ( RND ( 1 ) * 64 )
160 DRAW 2 AT X,Y : DRAW 2 AT X + 1, Y + 1
165 REM MAKE BUTTERFLIES PAUSE
170 FOR N= 1 TO 400 : NEXT N
180 XDRAW 2 AT X,Y : XDRAW 2 AT X + 1, Y + 1
190 GOTO 130
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7686
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 2,0,6,0,8,0,45,0,63,54,0
1040 NEXT LOC
1050 GOTO 20
```

Animated Conversations

Have you ever watched a Newsflash or Weather Alert travel across the bottom of your TV screen? Do you want to add this kind of action to your graphics? Just a few program lines will give you this form of animation.

Talk Lines

Let's use "Hello computer friends" as our moving message for the demonstration (Program 5.4).

Do you remember how HTAB and VTAB determine the horizontal and vertical locations of text? If HTAB T, then the FOR/NEXT sequence, T= 5 TO 50 will start writing our text at the left side of the screen and end it at the right. To avoid "trailing letters," the command CALL – 958 blanks the screen after the message is written at each horizontal position. See lines 20, 30, and 50.

In line 20, VTAB 20 places the message in the text window (bottom four lines of the screen).

The expression P= PEEK (–16336) in line 30 makes an attention-getting clicking sound.

That's all you need for Traveling Text, a five-line program.

Program 5.4 **Traveling Text**

```
5 REM PROGRAM 5.4 TRAVELING TEXT
10 HOME
20 FOR T= 5 TO 50 : VTAB 20 : CALL – 958
30 HTAB T : P= PEEK (–16336)
40 PRINT "HELLO COMPUTER FRIENDS"
50 NEXT T : GOTO 20
```

Note: This program can be used in conjunction with either the GR or HGR modes but will not function if the expression POKE –49234, 0 follows the commands GR or HGR.

Make Casso a Flasher

For a fun project, let's add "My name is Casso!" to Program 5.2 by making the following changes and additions:

```
445 FOR T= 45 TO 50 : VTAB : CALL – 958
450 HTAB T : P=PEEK(–16336 )
455 PRINT "MY NAME IS CASSO!"
460 NEXT T : GOTO 270
```

Traveling Is Broadening

I'm sure that when you see the text running along the screen under Casso you'll get some ideas about where else you'd like to try some Traveling Text. Go ahead, experiment! Pep up some graphs, add a little humor to a picture or two.

6

Creating a Slide Show

Easy Does It

Whether your interests are technical, business, or artistic, a slide show offers an efficient way of presenting your story in pictures. There are three kinds of slide shows you can use: computer slide shows, color photo slide shows, and video slide shows.

The principal needs for a computer slide show are an *assigned diskette* on which are stored the pictures you want to show and a program that will control the display of these pictures.

Get an Album

To get this project started, label an unused initialized diskette, "Slide Show #1" (or a name of your choice). You will use this diskette to store your pictures and the control program. Now let's see what it takes to store a picture.

Pix in the Fridge

There's nothing very complicated about storing a picture. You can do it in a few simple steps:

1. Choose the program that creates the picture you want and RUN it.
2. When the picture is complete, ENTER CTRL C to stop all program action.
3. Insert the "Slide Show #1" diskette in the controller.
4. Choose a name for your picture.
5. Very carefully, type in the following:

   ```
   BSAVE PICTURE NAME, A$2000, L$2000
   ```

   Note: You may not be able to see this text on the screen, but the computer will get the message anyway.
6. Press the RETURN key. The controller's red light will turn on, indicating that the picture data is being stored. When the light turns off you have stored a complete picture. Bravo!

The Checkout Counter

To be sure that everything functioned correctly, type in HGR, then type in BLOAD (PICTURE NAME). The picture will be loaded into Screen 1 and appear on the monitor.

Pictures Galore

Using the storage procedure described above, BSAVE the rest of the pictures that will comprise your slide show, storing them all on the new diskette. (A diskette will handle the control program plus about ten pictures.)

Getting Control

Now you need a control program that will find the pictures on the diskette and show them just like a slide projector.

Program 6.1 does it right. All you have to do is store your pictures on a diskette and substitute their names for those used in Program 6.1. ENTER the program on the assigned "Slide Show #1" diskette.

This program includes a neat trick that adds a professional touch to your slide show by ensuring that each picture is complete before it appears on the screen.

The command HGR2 in line 10 makes the monitor display Screen 2 (which is blank) while the computer is LOADing Picture 1 into Screen 1. When the picture is complete, POKE –16300,0 in line 30 switches the monitor display to Screen 1. Picture 1 then becomes visible and remains so until Picture 2 has been completely LOADed into Screen 2. Now POKE –16299,0 in line 50 switches the display to Screen 2 and Picture 2 becomes visible. Now that you get the idea, just keep going!

A reminder: In line 10 the expression D$ = " " means D$ = "CTRL D", but of course the "CTRL D" just leaves a blank space between the quotes. Now let's get on with the show!

Program 6.1 **Slide Show #1**

```
5 REM PROGRAM 6.1 SLIDE SHOW #1
7 REM HGR COMMAND CLEARS SCREEN 1
10 D$= " " : HGR : HGR2
15 REM LOAD PICTURE FOR SCREEN 1
20 PRINT D$ "BLOAD PICTURE 1"
25 REM "TURN ON" SCREEN 1
30 POKE -16300,0
35 REM LOAD PICTURE FOR SCREEN 2
40 PRINT D$ "BLOAD PICTURE 2, A$4000"
45 REM "TURN ON SCREEN 2"
50 POKE -16299,0
60 PRINT D$ "BLOAD PICTURE 3"
70 POKE -16300,0
80 PRINT D$ "BLOAD PICTURE 4, A$4000"
90 POKE -16299,0
100 PRINT D$ "BLOAD PICTURE 5"
110 POKE -16300,0
115 REM PROLONG DISPLAY TIME OF FINAL PICTURE
120 FOR N= 1 TO 2000 : NEXT N
125 REM REPEAT SEQUENCE
130 GOTO 20
```

The Real Thing

When you RUN Program 6.1, you will see a sequential display of your pictures in complete form. If you want to show more pictures, just continue the sequence of commands.

If you don't wish to repeat the picture sequence, eliminate the GOTO command in the last line of the program.

About A$4000

For future reference, it's worth emphasizing that even though a picture was *originally* stored at address A$2000, which is related to Screen 1, it can be loaded into address A$4000 for display on Screen 2.

Don't forget to add A$4000 after the name of each picture to be shown on Screen 2 as indicated in lines 40 and 80—including the comma!

Photographing Your Monitor Screen

Whatever the nature of your computer-generated images, photography offers a broad gamut of reproductions ranging from color slides and "instant" snapshot prints to poster-sized enlargements.

If your computer gear doesn't include a printer, don't overlook the use of photography for recording your programs. You can save yourself hours of boring effort by recording them on film instead of transcribing all those line numbers and commands from the screen to your notebook. (A lot cheaper than a printer, too.)

In the next few pages, you'll find "how to" information related to all of the above, first in a general way, and then right down to the "nitty-gritty." But first, let's review a few things you should know about the nature of computer-generated images.

Seeing the Subject Objectively

Computer-generated images are essentially the same as TV images. Each frame (the picture as a whole) is composed of 525 lines called scanning lines. The image you see on your screen may appear constant, but it is really a series of images displayed at the rate of 30 frames per second. Each frame is composed of two sections, or fields of odd- and even-numbered lines.

In a complete picture cycle, the odd-numbered lines are scanned in 1/60 second, and the even-numbered lines are then scanned at the same rate. The interlaced display of these two fields produces one complete frame of 1/30 second duration.

To photograph this complete picture cycle, you must use an exposure time of at least 1/30 second. If you use a shorter time, you will see a dark band across your picture.

If you use a camera with a focal plane shutter, you'll have to go to at least 1/15 second exposure because of the relationship between the shutter travel and the motion of the scanning beam in your monitor.

Ground Rules

Ground Rule 1: Don't ever even *think* of using flash when photographing your monitor screen. In addition to producing ruinous reflections, the flash will make the screen phosphors of the CRT glow with a resultant flattening of the image contrast. You'll have a completely washed-out picture.

Ground Rule 2: "Flare" (the glow around the edge of high-brightness lines and text) can ruin sharpness, so always reduce the monitor brightness to a bit below normal when photographing the screen. For consistency of results, put a mark on the brightness setting and reset the control to the same point for all picture taking.

Ground Rule 3: Avoid screen reflections. It's a good idea to turn off table lamps and ceiling lights to avoid unprogrammed images in your pictures. Watch out for reflections from windows, too. Now, let's look at some photographic options.

Go the Professional Route

If you have a market for your computer-generated images and a need for slides or prints in quantity, you can probably afford professional equipment to do the job. To see what's available, visit camera stores that handle audio-visual equipment.

I recommend the Kodak Instagraphic CRT Print Imager or the Kodak Instagraphic CRT Slide Imager. Both of these devices make it easy to get excellent pictures of your monitor screen without photographic expertise. As you can see in Figure 6.1, these imagers consist of a camera unit attached to a cone-like spacer that fits the monitor screen. Problems of tripods, focusing, and reflections are thus eliminated. Each imager includes a special lens that compensates for CRT screen curvature (minimizing image distortion). Cone spacers are available for CRT screens from 9 to 19 inches.

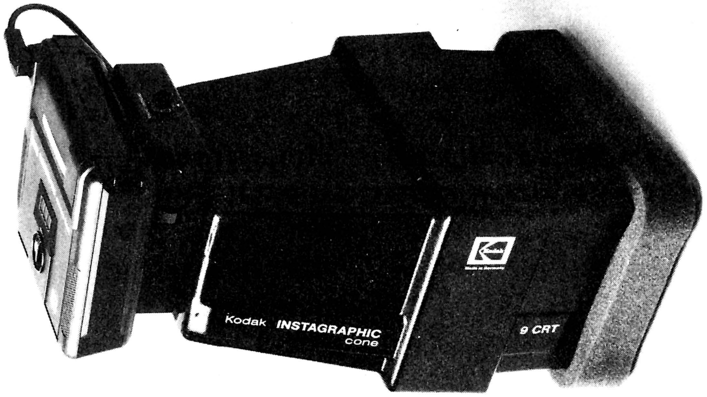


Figure 6.1 *Kodak Instagraphic CRT Print Imager*

Using the Kodak Instagraphic CRT Slide Imager with AVS 144-10 Kodak Instagraphic Color Slide Film, you can make one slide at a time.

Like Instant Print Film, development of this special slide film starts when it is exposed. Development time is about 12 minutes. After development, the transparency can be peeled away and will be dry enough to mount in a mounting device in about one or two minutes.

If you need prints, the Kodak Instagraphic CRT Print Imager functions in much the same manner. After the print is ejected from the imager, you can judge the exposure and color in about five to eight minutes.

While equipment of this general nature is not inexpensive, it is efficient and cost justifiable for professional use. Let's consider some other options.

The 35-mm Approach

For slide show presentations or in submitting artwork for commercial consideration, I recommend the use of a 35-mm single lens reflex camera. However, any good 35-mm camera can be used. Use a daylight-balanced medium- or high-speed film such as Kodak Ektachrome 200 Film or Kodak Ektachrome 400 Film. We'll get into exposure details later.

If you need prints in the snapshot to poster-size range, it's a good idea to use a fine grain 35-mm negative material such as Kodacolor VR 200 Film or Kodacolor VR 400. Of these two daylight-balanced films, the VR 200 is the sharper and preferable for large-scale enlargements. Both films have excellent exposure latitude.

For black-and-white prints a good starting material is Kodak Plus-X pan film. This is a fine grain film having an ASA speed of 125.

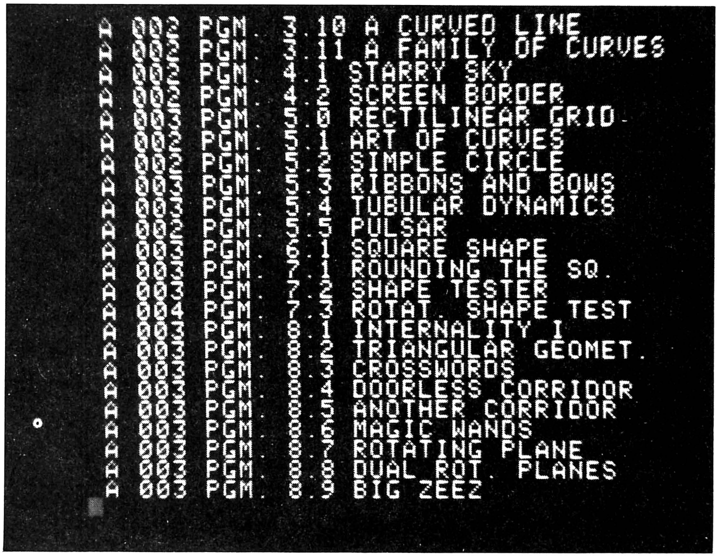
Going the Instant Route

For record purposes or "grab shots" to illustrate a point, you can get good results using one of the Kodak Instant or Polaroid cameras with the flash deactivated. If there isn't a control to turn off the flash, just remove the *flash* batteries.

For best results, use a tripod or some means of supporting the camera. *Measure* the distance from the lens to the screen.

The "Catalog" listing of programs shown in Figure 6.2 was photographed with a Kodak Instant camera using a close-up lens (+ 6 diopters). In case your photo dealer doesn't stock close-up lenses, there are two sources listed at the end of this chapter.

If you haven't time to use a tripod, just point and shoot with a "quick picture" camera. I've made hundreds of pictures satisfactory for notebook purposes in this fashion. (The exposure times used by these cameras are generally long enough to avoid the "dark band" problem.)



```
A 002 PGM. 3.10 A CURVED LINE
A 002 PGM. 3.11 A FAMILY OF CURVES
A 002 PGM. 4.1 STARRY SKY
A 002 PGM. 4.2 SCREEN BORDER
A 003 PGM. 5.0 RECTILINEAR GRID
A 002 PGM. 5.1 ART OF CURVES
A 002 PGM. 5.2 SIMPLE CIRCLE
A 003 PGM. 5.3 RIBBONS AND BOWS
A 003 PGM. 5.4 TUBULAR DYNAMICS
A 002 PGM. 5.5 PULSAR
A 003 PGM. 6.1 SQUARE SHAPE
A 003 PGM. 7.1 ROUNDING THE SQ.
A 003 PGM. 7.2 SHAPE TESTER
A 004 PGM. 7.3 ROTAT. SHAPE TEST
A 003 PGM. 8.1 INTERNALITY I
A 003 PGM. 8.2 TRIANGULAR GEOMET.
A 003 PGM. 8.3 CROSSWORDS
A 003 PGM. 8.4 DOORLESS CORRIDOR
A 003 PGM. 8.5 ANOTHER CORRIDOR
A 003 PGM. 8.6 MAGIC WANDS
A 003 PGM. 8.7 ROTATING PLANE
A 003 PGM. 8.8 DUAL ROT. PLANES
A 003 PGM. 8.9 BIG ZEEZ
```

Figure 6.2 ***Catalog Listing of Programs***

Serendipity 35

Any 35-mm camera that will focus down to two feet can be used to photograph entire screens of 12 inches or more diagonal width without the use of a close-up attachment.

If you have a nominal 12-inch monitor screen and a 50-mm lens on your camera, you'll find that the screen image will almost perfectly fill a 35-mm film frame at approximately 24 inches between the lens and the screen.

The Preliminaries

In the next few paragraphs it is assumed that you have your favorite 35-mm camera loaded with film and correctly positioned on a tripod in front of your monitor.

If you are using a viewfinder camera you can use a simple close-up framing device to align your camera with the screen as shown in Figure 6.3. All you need is a piece of cardboard the width of the screen with a pencil line drawn down the center.

An outstanding advantage of the single lens reflex camera is that "what you see is what you get." This feature makes it easy to be sure that you have filled the 35-mm frame with a sharp image.

The Nitty-Gritty

Assuming that you have a camera with provision for manual override, set your controls to the MANUAL position so that you can adjust the exposure values to those shown in Table 6.1 for the film and type of shutter you are using.

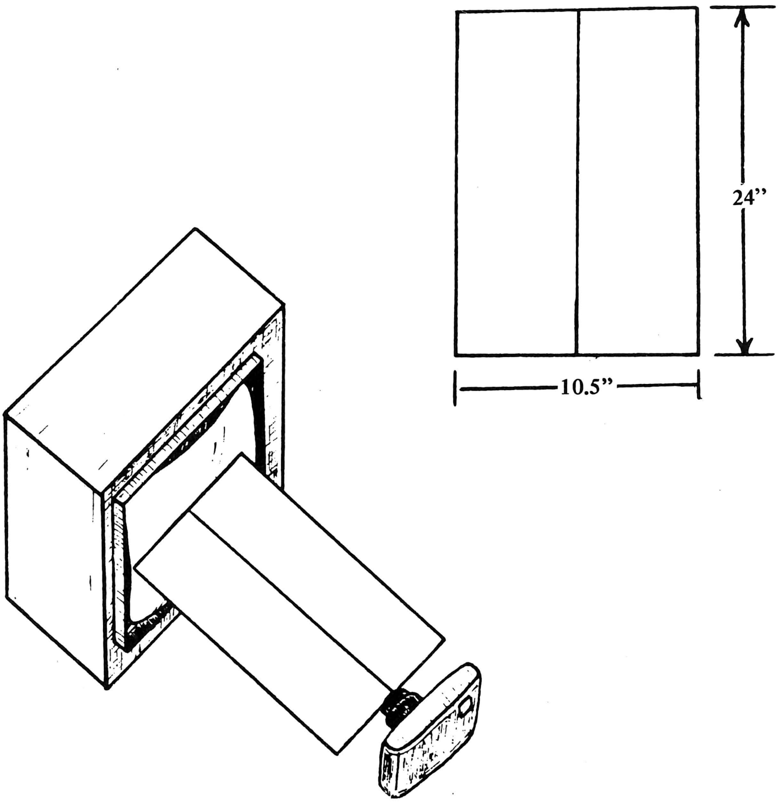


Figure 6.3 *Close-Up Framing Device*

If you are using an automatic camera without override, adjust the aperture as necessary to obtain a shutter speed of 1/30 second, 1/8 second with a focal plane shutter. Assuming that the lens openings are marked, you can now compare the lens and shutter values with those shown in Table 6.1. In this way you can predict your chances of success with the automatic camera.

You Press the Button

To establish a firm calibration for further work, you should run an exposure series over a range from two stops under the values indicated in Table 6.1 to two stops over. Make stepwise exposure increments of 1/2 stop per step for Ektachrome films. Because of the greater latitude of negative films, one-stop increments are satisfactory for Kodacolor films.

When these test exposures are processed, you'll have a good reference for future photographing sessions. Be sure to standardize your monitor brightness setting as suggested earlier.

Table 6.1 **Suggested Exposures for Monitor Photography**

Film Type	**Shutter Type**	
	Focal Plane	**Leaf**
Ektachrome 200 Kodacolor VR200	1/8 sec. at *f*/5.6	1/30 sec. at *f*/2.8
Ektachrome 400 Kodacolor VR400	1/8 sec. at *f*/8	1/30 sec. at *f*/4
Kodak Plus-X pan	1/8 sec. at *f*/5.6	1/30 sec. at *f*/2.8

A Filter Tip

You can get warmer tones in your slides by using an 85B or a CC 40R filter. When using either of these filters, increase the exposure by one stop.

These filters are not recommended for use with the Kodacolor films.

Further Information Department

Two excellent brochures that deal with photographing TV screens are available at your favorite camera store. Kodak Service Pamphlets AC-10, "Photographing Television Images," and KW-17, "Existing Light Photography," treat this subject in great detail. The KW-17 pamphlet contains very specific information related to monitor screen photography.

Lens Attachment Distributors

- Spiratone Inc.
  130 West 31st Street
  New York, NY 10001
- Tiffen Optical Co.
  90 Oser Avenue
  Hauppauge, NY 11787

Videotaping

By far the easiest approach to storing your Apple-generated images is by the use of a videocassette recorder, commonly known as a VCR.

If you want to put on a "really big show," with programs in motion as well as still pictures, a VCR can do it.

Videotaping dynamic programs and "stills" offers a compact, easily transportable means of supplying your material for review or display anywhere that a videotape player is available. (And no one can copy the code that creates your programs.)

Videotaping also offers the opportunity to add sound to your presentation as an aesthetic or purely communicative addition.

When you get a computer and a VCR together, you can really put your creative talents to work and have fun doing it. Add a color TV camera and some editing facilities, and the sky's the limit.

Let's start your computer/video experience by using a VCR to produce another form of a slide show.

The Line-Up

As is usually the case, it helps to get organized before you start assembling your video slide show. Here's how to do it.

First, make a *story-board* type of listing of the programs you're going to tape, in the sequence you desire. Note the display time for each picture or program.

Next, either put all of the programs you're going to tape on one disk or have the appropriate disks stacked in the sequence you want on the tape. This will make it faster and easier to do the taping.

Video

The only hardware you need to videotape your pictures is a good quality VCR, some tape, and a few "patch" cords. A second monitor is not a necessity, but it does make some operations more convenient.

Note: If you're anxious to use a TV camera for recording, feel free to skip ahead a bit. This section deals only with direct recording from the computer to the VCR.

What VCR?

To create a smooth-running presentation of your pictures, you need a videotape unit that is equipped with a PAUSE control.

The purpose of the PAUSE control is to eliminate picture loss or distortion between sequentially recorded segments. Check how this feature works before you buy a VCR.

Although there are many VCRs available that incorporate this feature, two that I know work very well are the RCA VFP 170 and the JVC HR 2200U. Both of these VCRs are of the VHS variety. Now let's assemble our equipment and get started.

Getting It Together

Let's start with the assumption that you are using a color TV set as a monitor.

1. Do not change the connection from your Apple computer to the TV set.
2. Connect the Video Output of the computer (phono connector at the right rear of the case) to the Video Input of the VCR.
3. Connect the RF Output of the VCR to the VHF Input of the TV.
4. To view what is being or has been recorded, change the TV channel switch from Channel 33 to Channel 3.
5. If you have a second monitor, you can avoid channel switching by connecting it as in step 3 above.

Wait One!

Just a little reminder to protect the innocent and prevent remorse: Before you push any buttons to activate your flight into the realm of computer video, make *sure* that the tape you're about to use doesn't have one of your favorite video recordings on it!

The Objective Is . . .

Now let's outline in a general way what you're going to do with this collection of hardware and software (before you do it).

1. RUN a program on the Apple and display it on the monitor.
2. Record the program on tape and display it while recording.
3. RUN a second program on the Apple.
4. Record this program.
5. Repeat these operations to the end of your story-board list.
6. Rewind the tape and view the complete sequence.

Nobody's Perfect

The first time you run through the sequence above, you'll probably have a few "flubs" attributable to pushing the wrong button, timing, or a miscued picture. Not to worry. Just rewind the tape and start over again.

Follow the instructions in the next paragraph while making a few practice runs before you try to record an extended series of pictures.

Some Practical Instructions

1. For the smoothest transition from one picture to the next, always put the VCR in the PLAY/RECORD mode at

the very start of the tape and immediately stop its action by pushing the PAUSE button.

2. Use the PLAY button to restart the recording (or see instructions for the VCR you're using).
3. After recording the desired footage, use the PAUSE button to stop recording.
4. Make sure that the next picture is showing on the monitor before restarting the recording in each sequence.
5. Use the PLAY button to restart the recording.
6. Observe the instructions for your VCR relative to any time limit on the use of the PAUSE button. On some units, the PLAY and RECORD buttons turn off automatically after about five minutes.

Following this sequence with a half a dozen pictures for a test run will give you a neat little slide show with no glitches, just one picture after another. New worlds to conquer—you're off and running.

Computergate

I wish that I could report that inserting pictures into an already recorded sequence is as easily and effectively done as the recording series just described. It isn't. Unfortunately, the PAUSE system doesn't prevent "picture rollover" when going from PLAY to RECORD. There are other picture disturbances when going from RECORD back to PLAY. What you're trying to do is really the beginning of what's called "editing," and you need editing equipment to do it.

Ye Editor

If your work justifies the time and expense involved in editing, you may find facilities that you can rent at a university or community college. Failing that, look for a commercial video lab to do the work.

Don't overlook the possibility of getting help from a cablevision outfit. You may be able to swap some video footage of your programs for the use of their editing gear. Smile, *your* graphics are on television!

The Not-Too-Candid Camera

Videotaping televised computer screen images has some interesting aspects. In the first place, the camera is designed for televising objects lighted with daylight or certain artificial illuminants. Its color sensitivity does not fit the color TV tube output that our eyes find acceptable. Net result: severe color distortion.

Depending upon the nature of your images and their intended use, this fact may or may not have great significance.

On the Plus Side

On the plus side, using a TV camera opens the door to virtually endless technique possibilities.

If you want to intersperse other types of images (for example, real objects ranging from artwork titles to machinery) with either your dynamic program pictures or "stills," you can handle these possibilities with a TV camera.

Practically all color TV cameras have a zoom lens. By using this feature, you can enlarge parts of your original screen image.

You can also create special effects by the use of filters or other optical devices. For example, using an auxiliary prismatic lens, you can surprise your viewers with multiple images. The multiple image trick is fun because it makes computer-oriented viewers wonder how you made the Apple do more than one thing at a time.

Give or Take

Summing it up, using a TV camera to videotape your computer-generated images opens the way to creative possibilities at the cost of color accuracy. Whether you gain or lose depends upon your needs.

Camera Action

Assuming that you have your program material ready, the preparation of a slide show using a TV camera essentially parallels the steps for direct videotaping. (The VCR doesn't care where the video originates.)

Note: For camera recordings you will need a second monitor. In the next few paragraphs, the computer-connected monitor is called Monitor 1, the VCR-connected monitor is called Monitor 2.

Camera techniques are your choice but here are a few hints:

1. LIST a program on the screen of your computer monitor.
2. With your TV camera "looking at" the monitor screen, adjust the lens so you can see the entire screen on Monitor 2.
3. Set the camera color balance control to the OUTDOOR position.

4. Momentarily turn the WHITE BALANCE control to the SET position. When the color on Monitor 2 becomes stable, return the switch to its normal position.
5. If there is a RED-BLUE color control, center it.
6. From this point on, make color adjustments on Monitor 2.
7. You are now ready to compose a program of color graphics using the TV camera.

Don't Just Stand There!

Put your creative talents to work with this flexible combination of equipment. Once you get started, you'll get more and more ideas to try. How about a video training tape? Get in the picture yourself. Sit next to Monitor 1 and let the camera capture you explaining what's on the screen. Why not?

Sound On

You'll never know how much fun it is to add sound to your computer graphics until you do it! With sound added, your graphics really come to life.

Adding narration to a technical or business-related slide show offers an opportunity to pep-up the presentation of facts. Appropriate sounds and/or music can also add a professional touch to your show.

Rhythm and Blues

For pure unadulterated *fun*, assemble a number of your dynamic programs on a videotape and add music that is rhythmically related to each program sequence.

Want to get started on a rewarding project? Here are a few suggestions. (*Warning:* The following project has been shown to be addictive and may cause loss of sleep due to unwillingness to quit!)

1. Select a series of graphics programs that vary in speed of operation.
2. As you review each program, think of what kind of music fits it: a waltz, martial band music, classical flute?
3. Make a videotape of these programs.
4. Assemble a group of records and/or tapes representing your first choice of music for each program.
5. Run the videotape and use your stereo equipment to play your chosen accompaniment. Fantastic!
6. By this time you have new ideas. Try them.
7. After you have some good matches between the program action and the music, use the AUDIO DUBBING feature of your VCR to record the music on your graphics program tape.

Be a Believer

If you feel that you are lacking the ability to select the music for this project, don't give up! Enlist a friend who has an ear for music and a room full of records to help. He or she will have a ball working with you. I guarantee that if you have as much fun as I did with my first graphics-with-sound videotape, it won't be your last.

Turning "Pro"

Today's TV commercials are loaded with computer-generated graphics made with special-purpose high-tech machines that are far beyond the monetary reach of a small studio. However, it is possible to produce low-cost commercials of excellent quality using a TV graphics generator with appropriate software, portable TV camera, VCR, and an Apple IIe computer. A small business handling TV commercials and videotapes of weddings, household inventories, and real estate items can easily justify the few-thousand-dollar cost of this equipment combination.

The key to mixing video and computer graphics is the TV graphics generator mentioned above. This device, with its software, performs all the magic necessary for overlaying graphics on video pictures—and more.

Pictured in Figure 6.4 is one of the best TV graphics generators available at this writing, the SYMTEC PGSIII. The illustration also shows a computer-generated graphic superimposed on a TV camera scene, as composed using the SYMTEC unit.

The PGSIII is an easy-to-use, low-cost video production tool with which you can create video still frames, animated sequences, weather maps, or interactive video training frames.

The unit has 512 x 480 resolution. Because of the very high resolution, only one screenful of information can be stored per 5½″ floppy disk. It has a palette of 4096 colors. Color substitutions can be made at any area of the screen. Customized software links the graphics generator with an Apple IIe computer to permit computer-generated overlays on camera video pictures. Standard software for the system requires the use of a graphics tablet (Apple or Kurta) and a 128K card dedicated to the system. Computer-graphics pictures stored in the usual fashion can be used with the PGSIII, but only by the use of custom software.

No doubt the future will bring integrated systems to permit an artist or designer easily to blend his or her video and computer graphics talents. But while we wait for this hi-tech compu-video marriage, a reasonably low-cost solution to mixing images of different parents is to add a sophisticated TV graphics generator to your Apple IIe.

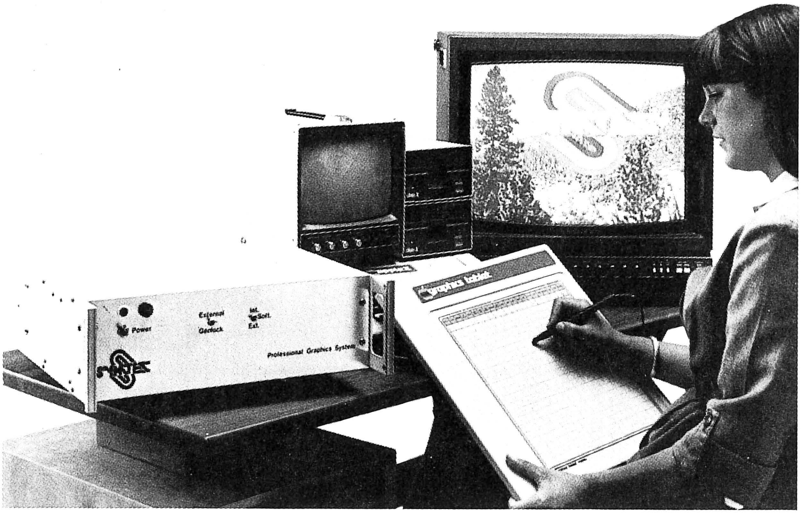


Figure 6.4 ***SYMTECH PGSIII Graphics Generator***

A Point of Departure

These first six chapters have described image creation by programming in the "original" Apple low- and high-resolution graphics modes. Now we've reached a turning point. The rest of this book will discuss and demonstrate what can be done with software and programming using a newer system called *double HiRes* and a new software animation package. Double HiRes, with its increased resolution and greater gamut of colors, brings the graphics capabilities of the Apple II computers to a much higher level of sophistication. Using the animation package, you can create objects and virtually make them come alive as they fill with color, turn, spin, and bounce on your computer screen.

To get you off to a good start with the newer system, the next chapter will introduce you to the terminology and conventions of double HiRes graphics software and programming. Double the resolution: double the fun!

Section 2

Choosing and Using Graphics Software

7

Introducing Software Terminology and Double HiRes

Software Talk

Among the many graphics software packages currently available, there are strong similarities in their capabilities and function. Sometimes the action is the same but the terminology differs. For this reason, some advance knowledge of terminology can be helpful as you pick your way through the orchards of Apple software.

The definitions and comments of this chapter deal with terms that are common to practically all Apple computer graphics software in use at this writing. Reading through these comments now may make it easier to understand graphics documentation in the following software demonstration chapters.

System Disks

All software packages operate with a *system disk*, which contains the programming necessary to make the system work.

Most system disks also include demonstration programs, fonts for text, and stored pictures.

Configuring the System Disk

Software writers expend great effort to maximize the variety of hardware combinations that any given software package can accommodate. However, for your computer/software combination to communicate intelligently within its total system, you must specify what elements are connected together. This is called "configuring the system disk" or "system setup." It's really not a complicated procedure.

The Nitty-Gritty of Configuration

Generally, the systems menu (the first menu to appear when you boot the disk) includes an item called "new configuration." When you choose this item, a series of queries on the screen will request that you enter such information as: the name of your computer, the number of disk drives, the kind of input devices, the name of your printer, etc.

Once you have supplied this information, it will not be necessary to configure the disk again unless you change some element of your system.

Font Disks

Some software packages include a separate disk in which type styles or fonts, icons, symbols, or even sample pictures are stored. Font disks featuring an extremely broad input gamut of type styles and icons are also available from Data Transforms and other software publishers.

Data Disks

It is general practice to store pictures created with any software package on a user-provided disk called a *data disk.* This disk has to be formatted in the operating system resident in the software package, either ProDOS or DOS 3.3.

Note: The details of making data disks are discussed later in this chapter.

Graphics Input Devices

Graphics input devices come in many forms, with what I consider an eclectic assortment of names: mouse; Koala Pad; Apple Graphics Tablet; paddles; trackballs; joystick; Ergopad. Yes, I know, they can't all have the same name—but take my advice and don't mention your mouse, Koala Pad, or whatever to your noncomputer friends. (They may begin to wonder if your elevator goes all the way to the top!)

These devices with the funny names are used to "draw," to "paint," to *control* a cursor position on the screen. In some software, they are also used to start, stop, or select an action or feature. In practical use, you'll find that software that makes provision for the use of graphics input devices can significantly speed up the rate at which you can create images or make menu selections.

How Do They Do It?

Electronic circuits measure the X and Y coordinates of a graphics input device's location and use the information to display the cursor position or to "draw" on the screen.

Menu-Driven Systems

Menu-driven systems are those in which software programming creates a series of *menus*, or listings of features and functions from which you make a choice by entering single letters that denote various features. The letters may be entered from the keyboard or by other input device such as a Mouse or a Touchpad, as described below.

Another form of menu-driven system is the kind used in Koala Technologies' "Koala Painter," "Graphics Exhibitor," and "Coloring Series II." The menus consist of boxes, within which options are listed. Adjacent boxes provide a space for a stylus-placed check mark. Choices are made by pressing a button on the Koala tablet. (The left button selects, the right button cancels.)

Windows

Software programming employing the "windows" approach was first introduced with the Apple Lisa computer. In this case, there are a series of features listed across the top of the screen. A mouse (or other device) is used first to select a feature, and then to pull down that feature's menu, from which you can select any of the items listed.

After a little experience, you'll realize that what's described in the previous paragraph can be done in much less time than it takes to read about it, so don't be intimidated!

"Dazzle Draw" is a good example of a graphics software package employing windows and pull-down menus.

Data Disk Details

Since the next few chapters will contain many references to the use of data disks, now's a good time to run through the process of making these disks for DOS 3.3 and ProDOS systems.

Any formatted DOS 3.3 disk can be used as a data disk. The "Hello" program built into the formatting procedure starts the disk operation as soon as the computer is turned on (or rebooted). For your convenience, the "Hello" program can call for a catalog of the disk. (This can be very useful if you don't have the disk's stored pictures listed on the disk label.)

Any formatted ProDOS disk can be used as a data disk if you have no need to boot up the disk. However, as noted below, three programs needed to provide communications between the computer and the disk drive must be added if you want to boot up the disk (for cataloging or other purposes).

Preparing a Data Disk for DOS 3.3 Operation

All that is required is to format a disk in the usual DOS 3.3 style. Insert a blank disk in D1. Then write a "Hello" program of three numbered lines as follows:

```
10 PRINT "DOS 3.3 DATA DISK FOR X-NAME SOFTWARE"
20 PRINT "TODAY'S DATE"
30 END
```

Then type in INIT HELLO. The disk will start, and the red light on the disk drive will turn on. When the red light goes out, the disk is formatted.

Preparing a Data Disk for ProDOS Operation

ProDOS formatted disks will not boot up and display a catalog unless they contain a "startup" program, the ProDOS file, and the BASIC.SYSTEM file. This means that when you make a ProDOS data disk, you have two choices: a disk that has maximum storage capacity but will not boot up, or a disk that will boot up but has somewhat less capacity.

Here are the alternatives:

1. Format a disk using the ProDOS system disk. Give the disk a volume name and label it "ProDOS data disk for X-Name Software."

   To find out (after storing some pictures) what's on this disk, you will have to use the catalog feature of the software in use.

2. Format a disk using the ProDOS system disk. Give the disk a volume name, then add copies of the system disk's ProDOS and BASIC.SYSTEM files, plus a startup file from *another* ProDOS disk. Do *not* copy the startup file from the ProDOS System Disk because it contains much memory-using data that you don't need. See Chapter 13 for a startup program and comments on why you might want to add the Beagle Graphics DHGR file, too.

For convenience, I recommend the second procedure until you have a great many pictures to store. I also recommend reviewing the description of "ProDOS Startup Disks" in Apple's book, *Basic Programming in ProDOS.*

Note: Both Dazzle Draw and the ProDOS version of Beagle Graphics make provision for cataloging ProDOS disks that *do not* contain the utility programs mentioned above.

Now that we have reviewed the general pros and cons of disk operating systems and the terminology of graphics software, it's time to move on to specifics. (Chapters 8–11 will acquaint you with four excellent graphics packages and demonstrate how they can be used creatively in a variety of projects.)

Double HiRes—A New Experience

The features of double HiRes graphics open the way to new levels of sophistication in the use of Apple computers for art and design.

With twice the resolution of normal HiRes and sixteen colors (plus mixed colors) to work with, the picture quality and design possibilities for these personal computers are greatly improved.

When double HiRes is incorporated into graphics software, exciting new features such as clip-and-paste, moving, inverting or "flipping" images, adding type to pictures, changing colors, and color-fill become available to broaden creative expression by the user. Many of these features are demonstrated in the chapters on Dazzle Draw, Fontrix, The Print Shop, and Beagle Graphics, just a few pages ahead.

Double HiRes also opens up new horizons for your creativity with a whole new set of commands that can be built right into your own programs. These commands are reviewed in the paragraphs below. (Double HiRes programming is discussed with many examples in Chapter 13.)

The Ampersand Commands

Programming in double HiRes graphics brings into play a set of commands known as *Ampersand commands* because they all begin with the ampersand (&) character.

The commands DRAW, HCOLOR, HGR, HPLOT, BLOAD, ROT, BSAVE, SCALE, TEXT, and XDRAW have ampersand equivalents. (For example, *& DRAW.)*

The commands & CIRCLE, & BOX, and & BCOLOR offer new and unique programming possibilities as described in Chapter 13.

In general, these commands are used just like their normal HiRes equivalents. Refer to the Beagle Graphics manual for command sequence requirements and some other ampersand variations.

Quotes and .AUX

The picture-saving and loading commands BSAVE and BLOAD also convert directly to & SAVE and & LOAD. Be sure to follow the terminology as described in the examples below:

- & SAVE "PICTURE", "PICTURE.AUX" will save a double HiRes picture to disk.
- & LOAD "PICTURE","PICTURE.AUX" will load a double HiRes picture. To display the picture, enter the following sequence of commands (with quotation marks required):

```
BRUN DHGR
& HGR2
& LOAD "PICTURE","PICTURE.AUX"
```

What Makes Double HiRes?

To enable the double HiRes commands to work, special software must be loaded into your computer. This software, which can be thought of as "a driver" (or sort of a mini-operating system), is currently available in two forms. First, as part of the Apple Computer Company's Extended 80 Column Text/Apple Color Card; and second, as the principal feature of the Beagle Brothers' Beagle Graphics software package.

The commands of these two systems are similar but not identical. The Beagle Graphics package has a larger number of commands, plus many routines for manipulating HiRes pictures and programs incorporated in its system disk. The Beagle Graphics version is relatively inexpensive and can be used with either an 80-column Apple IIe or an Apple IIc computer.

Now let's see how double HiRes images look via the software mentioned above. The next seven chapters explain the features of five software packages and demonstrate how easily they can be applied to some creative design projects. If you're anxious to use double HiRes in one of your programs, feel free to skip ahead to Chapter 13.

8

Picture It with Dazzle Draw

What Is It?

Dazzle Draw, a graphics software package produced by Broderbund Software, is one of the best graphics packages available for the Apple IIc/IIe computers. It consists of a system disk (which permits the making of one backup copy) and a user's manual. Images that you create using Dazzle Draw are stored on a separate disk called a data disk.

It's Versatile and Understandable

As you will see in the examples of Dazzle Draw operation, the features of this package are practically endless in number. Your creativity will be challenged, not restrained, by the selections available.

You don't need a computer science degree to understand the Dazzle Draw user's manual. It's written in plain language, there are good illustrations where needed, and there's an excellent glossary. In short, it's the best software user's manual I've ever seen.

What Can I Do with Dazzle Draw?

Here are some of the things you can do with Dazzle Draw:

1. Select any one of 16 colors.
2. Paint with solid colors or spray paint effects.
3. Operate in a zoom mode for accuracy of details.
4. Flip sections of a picture horizontally or vertically.
5. Create mirror images.
6. Copy and move images about the screen.
7. Create a slide show.
8. Create title slides or add titles to pictures.
9. Print your designs or pictures using a graphics-compatible printer.

These features and many more are accessed by the use of menus described and illustrated in paragraphs to follow.

Note: The Dazzle Draw system disk uses the ProDOS operating system described in Chapter 1. It functions in the double HiRes mode, providing 16 different colors and 560 horizontal plot points.

Setting Up the Dazzle Draw System Disk

Before you get started on your artistic efforts, put the Dazzle Draw disk in the disk drive and turn on your computer. After a title picture, you'll be greeted by a menu that asks questions about which input device (mouse, joystick, etc.), which printer, and which file system you'll be using. As indicated earlier, answering these questions is called "setting up the system disk." You have to deal with this routine only once, unless you make a change in your system.

The Mouse Does All the Work

By using the mouse/cursor technique (to be described shortly) for selecting menus and features, Dazzle Draw makes it fast and easy to create your designs or drawings. A pull-down menu approach is used for selection of operating features.

There are only a few cases where you have to enter any information by typing. After a little experience with this package, you'll be amazed at how quickly you can create any form of image within its broad range of capabilities.

Putting Dazzle Draw to Work

The best way to see how easy it is to use Dazzle Draw is actually to run the software, or at least see a store demonstration. However, if you can't get your hands on this nifty package, just follow the steps of the Dazzle Draw design project described and illustrated in the following paragraphs. You'll find that just reading about what you need to do to make a picture will help you visualize how this system can enhance the graphics features of your Aple IIc/IIe and extend your own artistic capabilities.

Don't be put off by the many steps involved—using the mouse and Dazzle Draw's windows make them go very fast.

What's Needed

First, let's see what's needed. Dazzle Draw assumes that you have an Apple IIc or a 128K Apple IIe with the Extended 80-Column Text/Apple Color Card, a disk drive, color TV or monitor, and one of the following:

An Apple mouse
An Apple Graphics Tablet
Other drawing pad such as Koala or ErgoPad
Animation Station
A joystick

A Minor Assumption

The Dazzle Draw user's manual assumes that you will be using a mouse to move the cursor about the screen, but instructions are included for the use of graphics tablets, drawing pads, and joysticks. In my experience, the Apple mouse has excellent articulation (relationship between physical movement and screen display position), so I recommend it over a joystick or drawing pad.

It's Mouse Aerobic Time

In the description that follows, it is assumed that you are using a computer equipped with an Apple mouse or its equivalent.

Figure 8.1 shows how the Dazzle Draw screen is divided, with five pull-down menus at the top, a center section for your

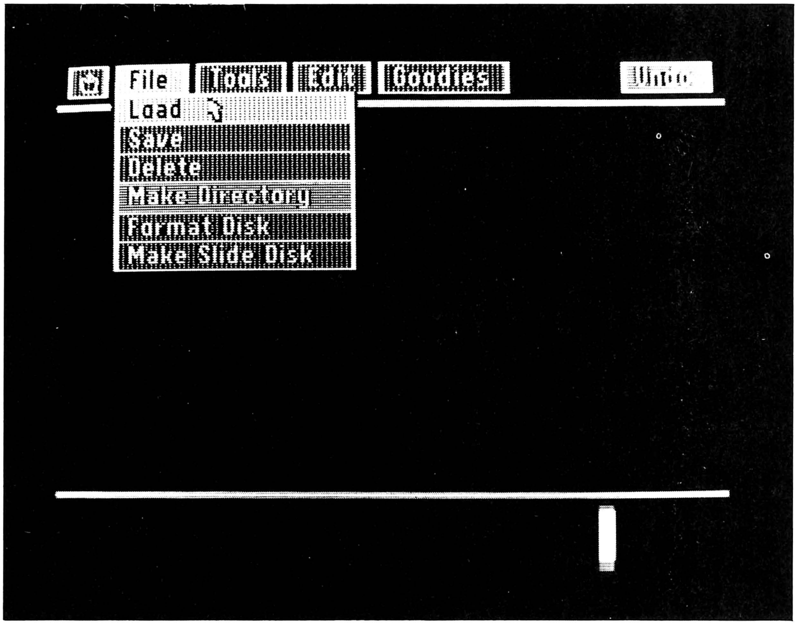

File
Tools
Edit
Goodies
Load
Save
Delete
Make Directory
Format Disk
Make Slide Disk

Figure 8.1 ***Dazzle Draw Main Menu***

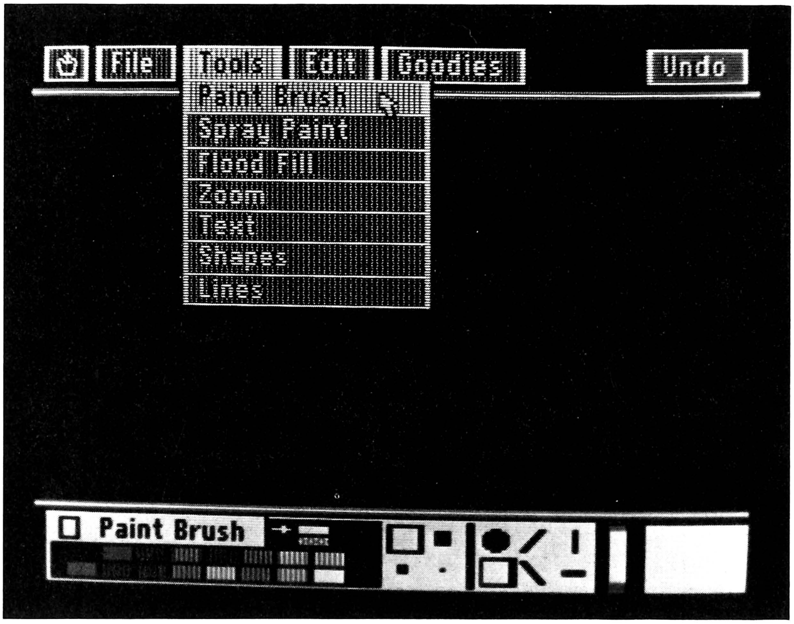

File
Tools
Edit
Goodies
Undo
Paint Brush
Spray Paint
Flood Fill
Zoom
Text
Shapes
Lines
Paint Brush

Figure 8.2 ***Dazzle Draw Paint Brush Menu***

drawings or paintings, and a bottom section for selection of "tools" and colors.

Figure 8.2 shows how to choose a menu or select a feature listed on the menu using a mouse. Position the mouse arrow/cursor over the desired item and push the button on the mouse. Then, holding the button down, move the mouse as needed to "pull down" the menu and highlight your selection. Now let up on the mouse button and you're ready to use the feature of your choice.

You Need a Data Disk to Store Your Pictures

The first feature you're going to use is FORMAT. Here's how you do it. (More details in Chapter 7.)

Boot up the Dazzle Draw system disk and when the menu line appears on the screen, *remove* the Dazzle Draw system disk and insert a blank disk in its place. Pull down the FILE menu and select FORMAT. Click the mouse button and the disk drive will start the format operation. When the disk drive red light goes out, you have a data disk and you're ready to begin your artistic efforts. Leave this disk in the drive.

Design a General Form for Cards or Letterheads

Figure 8.3 shows a general form that can be customized in a variety of ways.

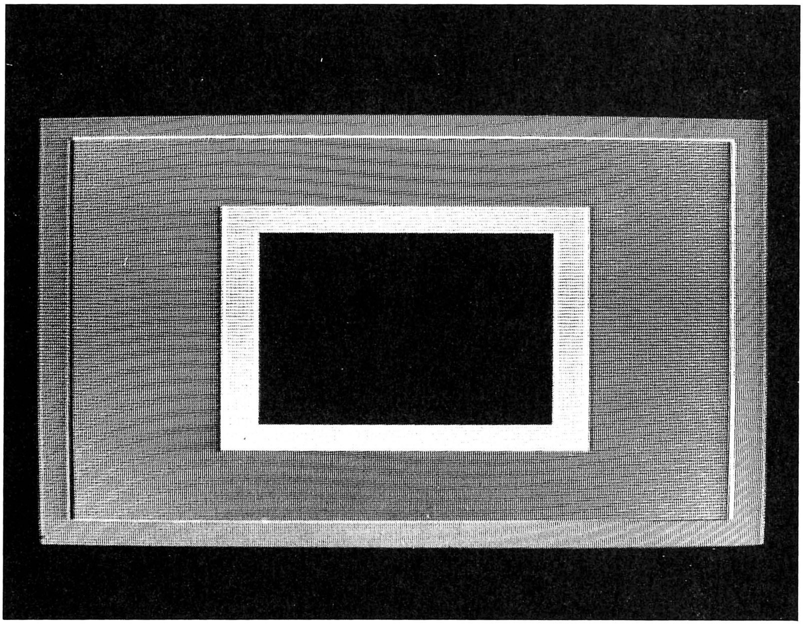


Figure 8.3 *A General Form*

Here are the steps necessary to create this basic form.

1. Pull down the TOOLS menu and select PAINTBRUSH. Use the mouse to select the largest round brush. Click the mouse on the red color bar.
2. Paint the outside border by "dragging" the mouse from left to right and top to bottom around the screen edges. (Dazzle Draw has a nifty feature that applies here: To get absolutely straight lines, hold down the Open Apple button as you paint each line. Clever!)
3. Select the color white and repeat the border painting steps to create an inner border.
4. Pull down the TOOLS menu and select FLOOD FILL and the color blue for the area between the inner and outer borders.
5. Position the mouse arrow in the area between the borders and click the button. The area will fill with the blue color selected above. Now you have a *blank form* that can be customized as desired. Let's SAVE this blank form so we'll have it available for customizing and avoid having to start from scratch in any future use.
6. The SAVE operation goes like this: Pull down the FILE menu and select SAVE. Indicate that you want to SAVE a picture named "Cardform." Click the mouse over the word OKAY at the right of the screen and your card design will be saved.
7. Cancel FLOOD FILL by clicking the mouse over the little FLOOD FILL square icon at the lower left of your screen.
8. Pull down the TOOLS menu again and select TEXT.
9. Select a font size and style by clicking the mouse over these items. (They'll change each time you click the button.)

10. You can insert your name or other text in the inner rectangle of the blank form. You will have to "eyeball" the type size and number of words according to the available space. Use the mouse to determine the location of the first letter on any line. The following letters are positioned automatically as you type.

Assuming that you saved the blank card after Item 5 above, you now have a blank to use at the start of other designs, such as shown in Figure 8.4. The design around the edge of the card in this illustration was created by using Dazzle Draw's shapes feature.

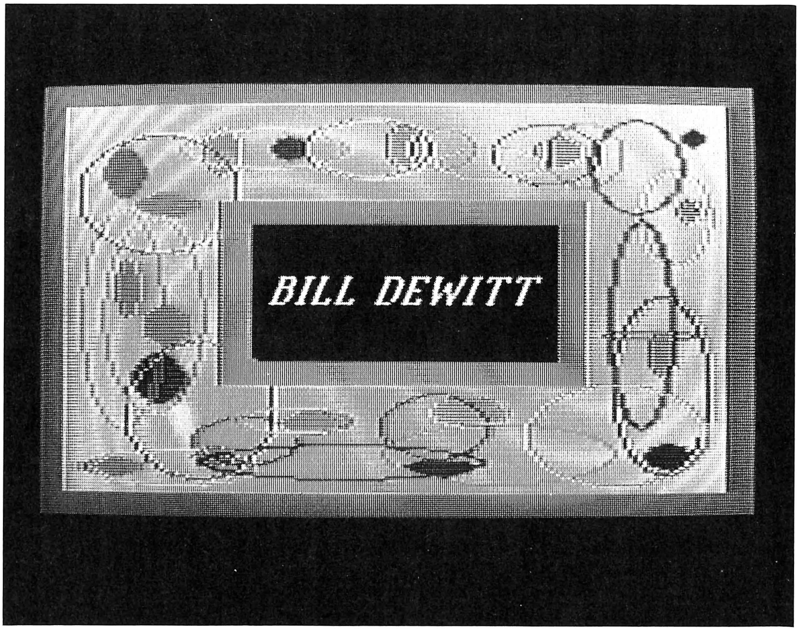


Figure 8.4 ***Dazzle Draw Shapes Design***

Using Dazzle Draw's cut-and-paste provides another way to create a design. In Figure 8.5 (see color illustrations following page 146) the design around the edge of the card was created by "cutting" a small area out of a picture (created with Beagle Graphics—then converted to Dazzle Draw format) and "pasting" it around the edge of the card.

Project Complete

This little project demonstrates how you can create a useful design in ten major steps with the Dazzle Draw package. The menu system gives you a fast-operating point-and-click system that's hard to beat.

More Dazzle Draw Pix

The versatility of Dazzle Draw is illustrated further with five additional pictures (turn to the color illustrations following page 146 to see Figures 8.6–8.10). Figure 8.6, "A Solar Sorority," was created by using Dazzle Draw's shapes feature. Figures 8.7, 8.8, 8.9, and 8.10 were "painted" using the mouse and Dazzle Draw's paintbrush with a variety of colors and brush sizes. The dual images of "The Kid" in Figure 8.8 were produced by using two other Dazzle Draw features, capture and flip. "The Kid in a Red Cap" pictured in Figure 8.9 was derived from Figure 8.8 by altering colors with the exchange colors feature. These examples will give you an idea of what you can do in minutes with this excellent software package.

Dazzle to Go?

To expand the possibilities of Dazzle Draw and the Beagle Brothers Graphics package (described in Chapter 11), you can make use of two programs listed in Chapter 12 that simplify the task of converting Dazzle Draw pictures to the Beagle Brothers Graphics system, and vice versa. These programs make it possible to build a library of pictures created in either of the two systems.

9

Fontrix Is Your Type

What Is It?

Fontrix Version 1.5, produced by Data Transforms, is another highly rated graphics package available for Apple computers. The package consists of a system disk, another disk in which 11 standard fonts reside, and a user's manual.

Hardware requirements are an Apple II+, IIc, IIe, or III (in emulation mode), 48K RAM, and a DOS 3.3 operating system.

As is the case with all similar software, graphics you create using Fontrix must be stored on a data disk.

Data Transforms encourages you to make backups of the system and font disks.

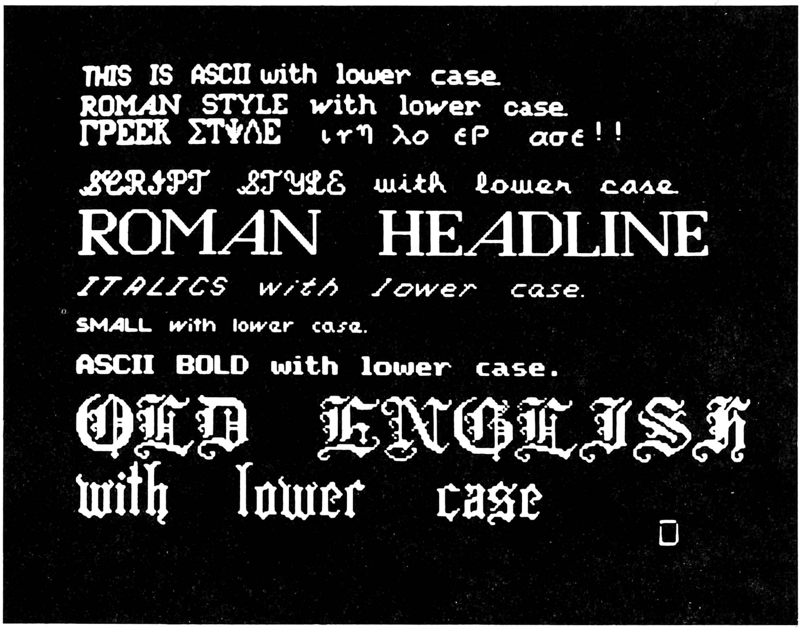
Graphics Input Devices

Graphics input devices that will work with Fontrix include the following: graphics tablets made by Apple, Talos, and Koala (Touch Pad); joysticks made by Apple, Hayes, Kraft, and TG Products; mouse units by Apple and Product Associates; and paddles made by Apple, Hayes, Kraft, and TG Products.

There are over 40 printers that can be used with Fontrix using appropriate interface cards where necessary.

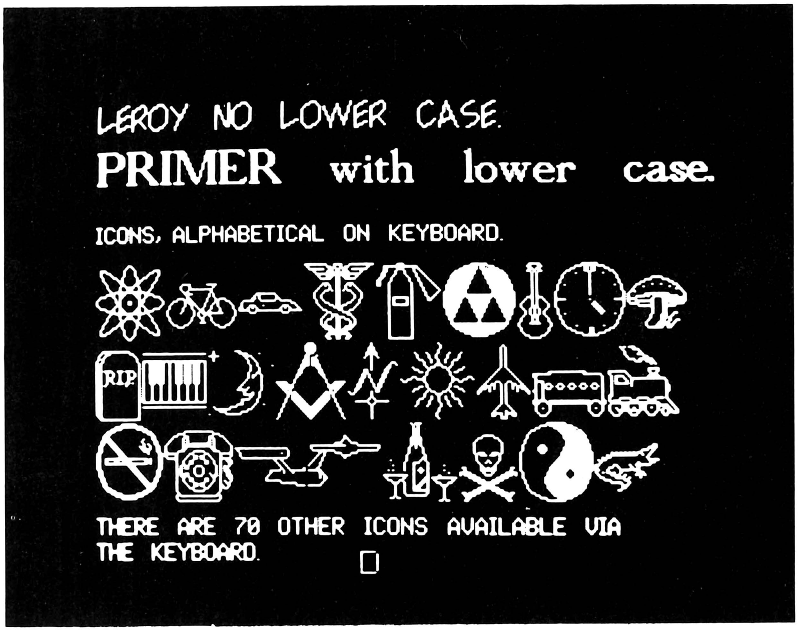
Which Tricks with Fontrix?

The best applications of Fontrix appear to be in making posters and newsletters, but the amazing gamut of typefaces and symbols available make it useful for almost any kind of printed matter. See Figures 9.1, 9.2, and 9.3.



THIS IS ASCII with lower case
ROMAN STYLE with lower case
ΓΡΕΕΚ ΣΤΨΛΕ ιτη λο εΡ ασε ! !
SCRIPT STYLE with lower case
ROMAN HEADLINE
ITALICS with lower case.
SMALL with lower case.
ASCII BOLD with lower case.
OLD ENGLISH
with lower case

Figure 9.1 *Fontrix Font Styles*



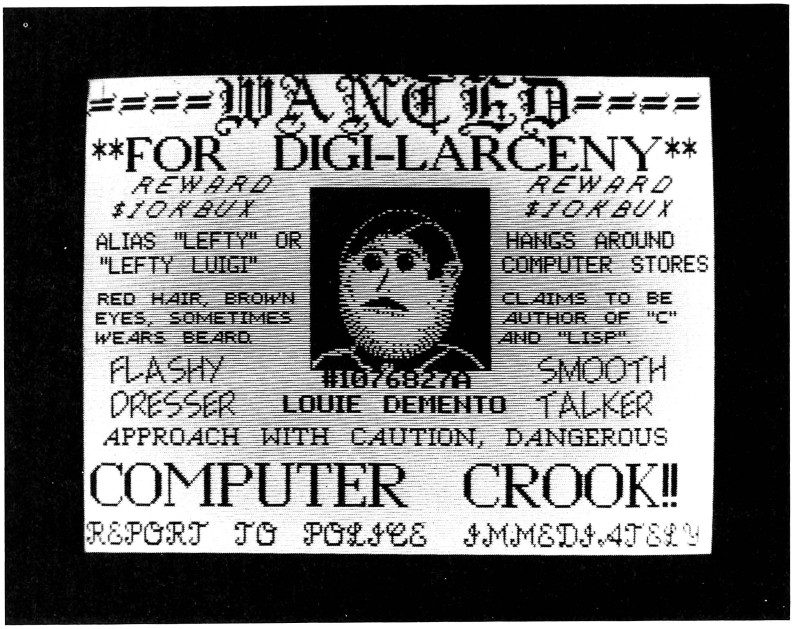
LEROY NO LOWER CASE.

PRIMER with lower case.

ICONS, ALPHABETICAL ON KEYBOARD.

THERE ARE 70 OTHER ICONS AVAILABLE VIA THE KEYBOARD.

Figure 9.2 *Fontrix Icons*



====WANTED====

FOR DIGI-LARCENY

REWARD $10K BUX

REWARD $10K BUX

ALIAS "LEFTY" OR "LEFTY LUIGI"

HANGS AROUND COMPUTER STORES

RED HAIR, BROWN EYES, SOMETIMES WEARS BEARD

CLAIMS TO BE AUTHOR OF "C" AND "LISP"

FLASHY DRESSER

#1076827A

LOUIE DEMENTO

SMOOTH TALKER

APPROACH WITH CAUTION, DANGEROUS

COMPUTER CROOK!!

REPORT TO POLICE IMMEDIATELY

Figure 9.3 *Fontrix Font Demonstration*

More About the Fontrix Package

Fontrix is a rather sophisticated package that calls for much more of a learning period than Dazzle Draw or Print Shop, but the wide range of capabilities you acquire in using it can make the effort well worthwhile.

To quote the Fontrix manual, "Fontrix is actually quite a few things: It's a character generator, a drawing board, a color painting system, a typesetting device, a means of combining the printed word with oodles of graphics on a screen large enough to fill an entire printed page . . . it's more than that even . . . it's an exercise in imagination. Create a use to suit your needs." After that modest statement, let's take a closer look.

It's Fully Packed

In addition to the 11 fonts that are included in the software package, many others are available on Fontpak character-set disks, which are identified by volume numbers. If you have very special needs, you can create your own type styles with one of Fontrix's many built-in features.

You can use these fonts to create printed pages or to add text to high-resolution graphics pictures.

An outstanding Fontrix feature called Graffiles enables you to make and print graphics up to 16 times larger than the Apple screen. The hard copy printout size is limited by the printer characteristics. (This is really a great feature! More later.)

Menu Time

Fontrix software is menu-driven. This means that there is a menu for each major feature. Your choice from any menu is made by entering letters or numbers from the keyboard. (The graphics input devices mentioned above cannot be used for this purpose.)

About the User's Manual

The Fontrix manual is a hefty, comprehensive, good quality notebook–style package. All Fontrix features are described in detail with specific examples. Illustrations show all of the menus as you would see them on the monitor screen. The looseleaf format makes it easy to make update changes. Data Transforms make updates available at very reasonable prices.

Add a Title to a Picture

Hands-on experience is hard to beat, but you can get a feeling for the way Fontrix Version 1.5 works by following the steps necessary to put a text or description on a graphic as described in the following paragraphs. Figure 9.4 shows a crisp title added to a picture, a good project for Fontrix's talents.

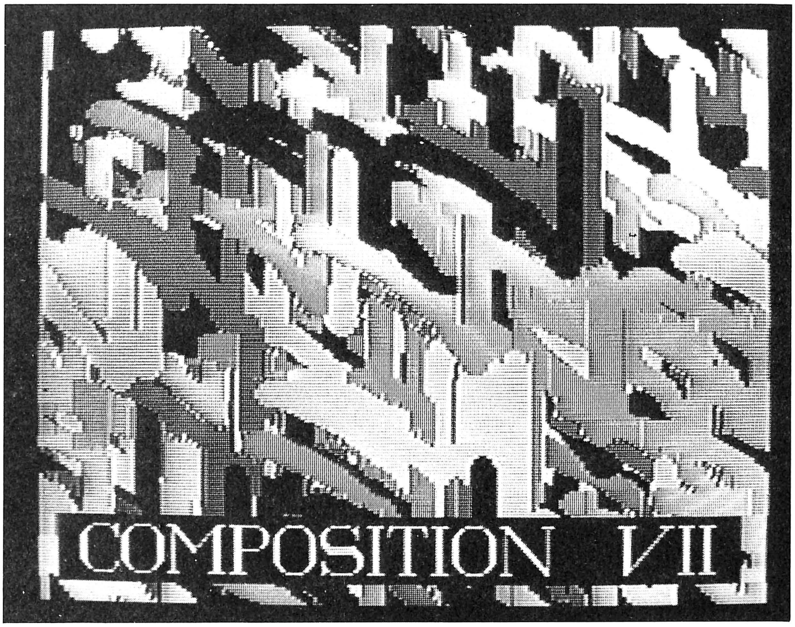


Figure 9.4 ***Text Added to Picture***

Remember, this is just one of the many ways that you can use Fontrix to extend the usefulness of your computer and stretch your imagination!

What's Needed

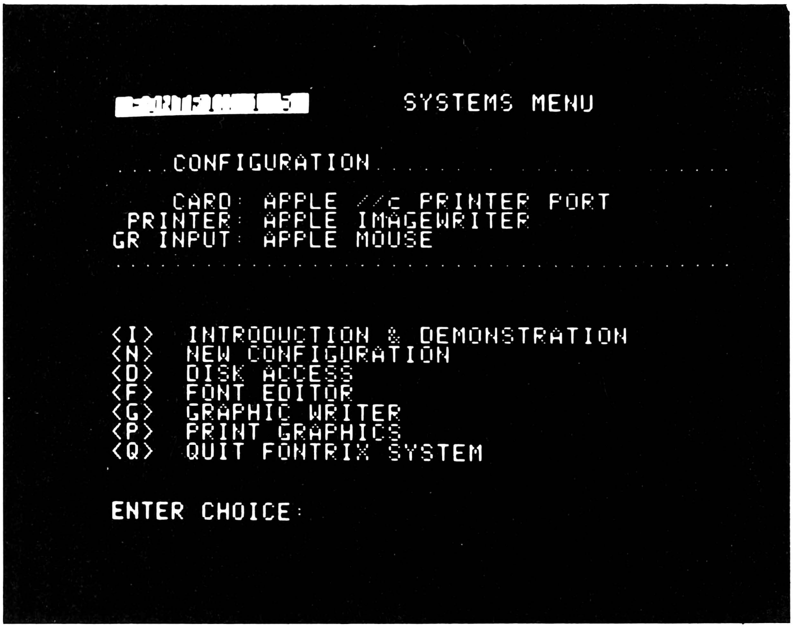
Fontrix software assumes that you have a monitor, some combination of the computers and graphics input devices mentioned earlier, the Fontrix Version 1.5 font disk, and a formatted data disk. The Fontrix system disk must be configured to your system as described earlier.

Starting Up

When you boot up the Fontrix system disk, you'll be greeted with the Systems menu shown in Figure 9.5. Enter G to access the Graphic Writer program.

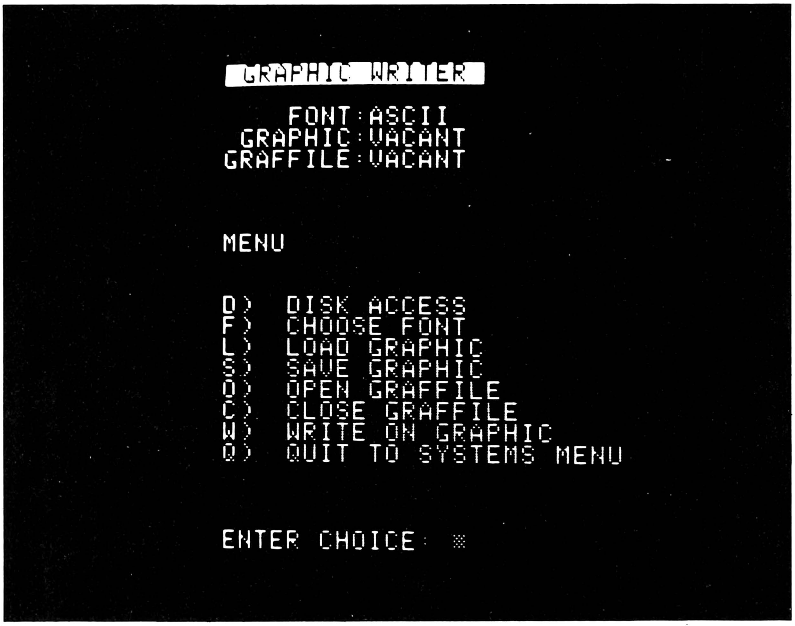
Now you're ready to make a series of menu choices and disk changes as listed below.

1. *Choosing a font:* When the Graphic Writer menu shown in Figure 9.6 appears, enter F to choose a font. Let's select a large-size type style, one called "Roman Headline." Replace the system disk with the font disk. Enter the font name and it will be loaded into the system.
2. *Loading a picture:* When the Graphic Writer menu reappears, enter L to load a picture stored previously on a data disk (which we'll call the "picture disk").
3. Remove the font disk and replace it with the picture disk.
4. Enter the name of the picture. We'll use the abstract "painting" called "composition VII" which is stored on



```
FONTRIX 1.5              SYSTEMS MENU

    ....CONFIGURATION...............................

         CARD: APPLE //c PRINTER PORT
      PRINTER: APPLE IMAGEWRITER
     GR INPUT: APPLE MOUSE
    ................................................


<I>   INTRODUCTION & DEMONSTRATION
<N>   NEW CONFIGURATION
<D>   DISK ACCESS
<F>   FONT EDITOR
<G>   GRAPHIC WRITER
<P>   PRINT GRAPHICS
<Q>   QUIT FONTRIX SYSTEM

ENTER CHOICE:
```

Figure 9.5 *Fontrix Systems Menu*



```
GRAPHIC WRITER

    FONT:ASCII
 GRAPHIC:VACANT
GRAFFILE:VACANT


MENU

D)  DISK ACCESS
F)  CHOOSE FONT
L)  LOAD GRAPHIC
S)  SAVE GRAPHIC
O)  OPEN GRAFFILE
C)  CLOSE GRAFFILE
W)  WRITE ON GRAPHIC
Q)  QUIT TO SYSTEMS MENU


ENTER CHOICE:
```

Figure 9.6 *Graphic Writer Menu*

the companion disk for this book. After the disk drive has loaded the picture data into the screen memory, the Graphic Writer menu will reappear.

5. *Enabling the write function:* Enter W to select "Write On Graphic."

6. The Graphic Writer Command Keys Table will appear, displaying 26 "CTRL (A through Z)" commands.

   Note: Although this is an extremely useful table, you will be using only three of its listed commands for this project. They are as follows:
   a. Enter CTRL A if you forget which command keys you need. CTRL A makes the Command Key Table reappear.
   b. Enter CTRL G to go from keyboard control of the cursor to the mouse. Enter CTRL G again to go back to the keyboard. This is called *toggle* action.
   c. Enter CTRL Q to return at any time to the Graphic Writer menu.

7. To view the previously loaded picture, press any key. The Command Keys Table will disappear and the picture "Composition VII" will appear.

   Remember that in step 5 we enabled the text writing function. Now to help us write text, a rectangular cursor approximately the size of the Roman Headline typeface has appeared in the upper left corner of the screen.

8. Enter CTRL G to activate the mouse and use it to position this pulsing cursor in the lower left of the picture as shown in Figure 9.7. Enter CTRL G again to return control of the cursor to the arrows and space bar.

9. *Creating a black background:* Now use the cursor and the left and right arrows to "eyeball" the location of a solid black background for the white text picture title.

10. Pressing the space bar blanks out picture details and leaves a black screen.

After you have found where to start and end the black background, use the space bar to create it.

11. Repeat step 10 to increase the size of the black area on which the text will be printed.
12. Enter CTRL G and locate the cursor near the left end of the black area. Enter CTRL G again so that control of the text location returns to the keyboard.
13. *Add the title:* Now enter the picture name "Composition VII" and the project is complete! Refer back to Figure 9.4. How about saving the titled picture?

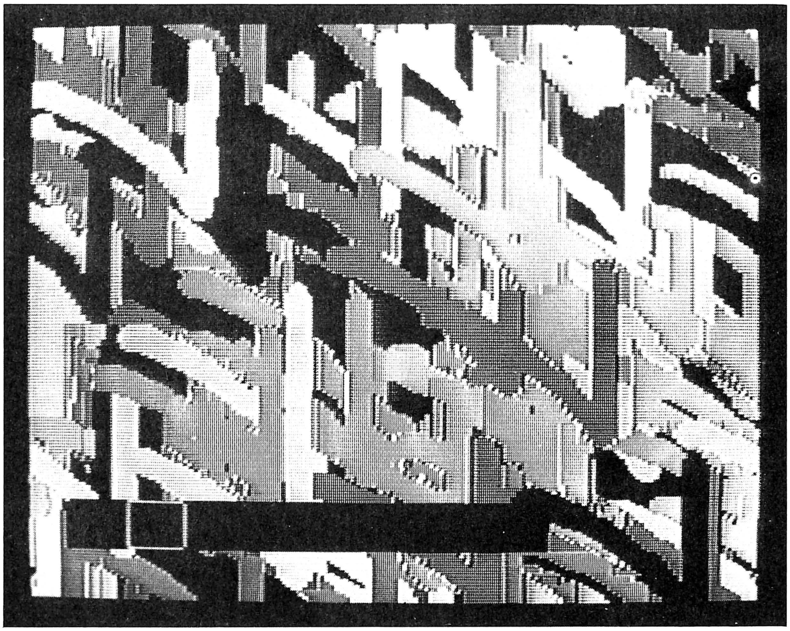


Figure 9.7 *Creating a Background for Text*

Save Composition VII?

Saving this titled picture for future reference is easily done. First, make sure that you have a DOS 3.3 initialized disk in place, then:

1. Enter CTRL Q to get back to the Graphic Writer menu.
2. Enter S to SAVE the picture.
3. Enter the name of your choice and, "By George, you've got it!"

10

Everyman's Print Shop

What's a Print Shop?

The Print Shop is a graphics software package produced by Broderbund Software. It's probably the best package for its purpose available for Apple computers. It consists of a system disk and a reference manual. Also included is a handy reference card which lists text entry commands and shows the type styles, numbered graphics, and patterns that are stored in the system disk. This means that everything you need to start designing any printed matter is literally at your fingertips.

Images that you create using the Print Shop must be stored on a DOS 3.3 formatted data disk, which you supply.

In keeping with Broderbund's practice, one backup system disk can be made from the original.

Print Anything—Well, Practically Anything

Like Broderbund's Dazzle Draw, the features of the Print Shop are practically endless. Its system disk is loaded with everything you need to design and print out greeting cards, invitations, stationery, letterheads, banners (big, long banners with large letters), and signs. If you have pictures or charts stored on a data disk, the Print Shop can be used to add text to them. This is a nifty way to make signs or mini-posters.

Remember George Eastman?

Although the main thrust of the Print Shop documentation is understandably directed at paper printers, users who need big prints should remember that an easy-to-make photograph of the screen can be used to get an inexpensive "blow-up" print in black and white or color. (See Chapter 6 for more information on screen photography.)

The Manual's Good, Too

The Print Shop reference manual says, "The Print Shop is an easy-to-use program that lets you view, select and assemble the various elements of a personalized, decorative message, and print it on your regular computer paper, all in a matter of minutes. Creative expression and practical computer-generated graphics have never been this easy, foolproof, or so much fun!"

The manual also indicates that you really don't need to read it to use the Print Shop software. "Just follow the prompts," it says!

My own experience with the Print Shop certainly supports both of these statements.

So far as the manual itself is concerned, I would rate it excellent. It is complete, logically organized, well written, and fully illustrated. The manual covers not only the software operation but also includes tips about printer operation.

Setting Up the Print Shop System Disk

The first time you start up the program, the word "setup" will be highlighted on the main menu. (Setup is sometimes known as "configuring the system disk.") Here's how it goes.

Press RETURN and you will be asked to specify (in sequence): printer make, printer interface card, printer interface card slot, and number of disk drives.

After you supply this information, the printer is all ready to run a self-test. Make sure your printer is loaded with paper and ready to run. Press RETURN. The printer will respond by printing a one-line greeting in double-width letters. If the printer doesn't respond, press ESC and go over the steps listed above.

You won't need to set up or configure the system disk again unless you change some of the elements in your system.

A la Carte

The Print Shop main menu features options (such as greeting card, letterhead, sign, etc.) listed in large-size text alongside colorful snapshot-size option-indicating pictures that change as options are chosen. This is a clever touch that makes most other menus seem pretty dull. You'll find information on how to choose and use decorative graphics later in this chapter.

Options can be selected by using the Apple UP and DOWN arrows and pressing RETURN when your choice is highlighted. You can use the Touch Pad or Koala Pad mode of selection, but you will find the keyboard method is very fast in operation.

Outstanding in the Field

An award-worthy feature of the Print Shop's programs is that by using the ESC button you can back up from where you are to correct an entry (or whatever the reason), in practically any case but the loading of a picture from another disk. If you have ever used software that sends you back to square one to correct any error, you're going to love this back-up capability.

What's Needed to Use Print Shop

To put the Print Shop to work you need: Apple II+/IIc/IIe with at least 48K memory, or an Apple III in emulation mode; a disk drive; a monitor or TV set; one of a recommended list of printers with an interface card if needed. Optional items are: a joystick or Koala Pad, a data disk, pinfeed paper, and/or colored paper and envelopes.

The System Disk

The Print Shop's system disk is loaded with everything you need to design and print out greeting cards, invitations, stationery, letterheads, banners, and signs. In addition, the software enormously expands the creative possibilities for the creative user by permitting the use of pictures or charts stored on DOS 3.3 data disks.

About Data Disks

Making a data disk is a snap. See Chapters 7 and 13.

To get organized for a little action, let's assume that you have two disk drives with the system disk in Drive 1, a DOS 3.3 data disk in Drive 2.

Now we're ready to walk our way through a Print Shop project so you can see how easily you can create the artwork for a book cover or a report.

It's Fast

If you have sneaked a peek ahead, you may be a bit concerned about the number of listed steps from start to finish of this project. *Don't worry!* It all goes very fast. The Print Shop's easy-to-use menus and instructions get you to the printing stage in a matter of minutes.

Design a Book Cover

Here's what we're going to do: design an attractive, eye-catching book cover. The entire cover will be filled with an attractive abstract design in saturated colors. We'll use a rather unusual font and do the title in white-outlined black letters. (The Print Shop cannot originate color designs, but it can overlay text on color images created with other software or by graphics programming.)

The Design Steps

First, a reminder to make life easier for you: All of the Print Shop menus include a *highlighting bar,* which you position to signify your choice of action. The system is really quite easy to use. Just use the UP and DOWN arrows (or a graphics input device) to make a choice and then press RETURN.

Remember, if you make a mistake or change your mind, just keep pressing ESC to back up as far as you wish.

1. After booting up the system disk, choose "Screen Magic" from the main menu. (See Figure 10.1.)

2. From the "Select Activity" menu, choose "Get Screen."

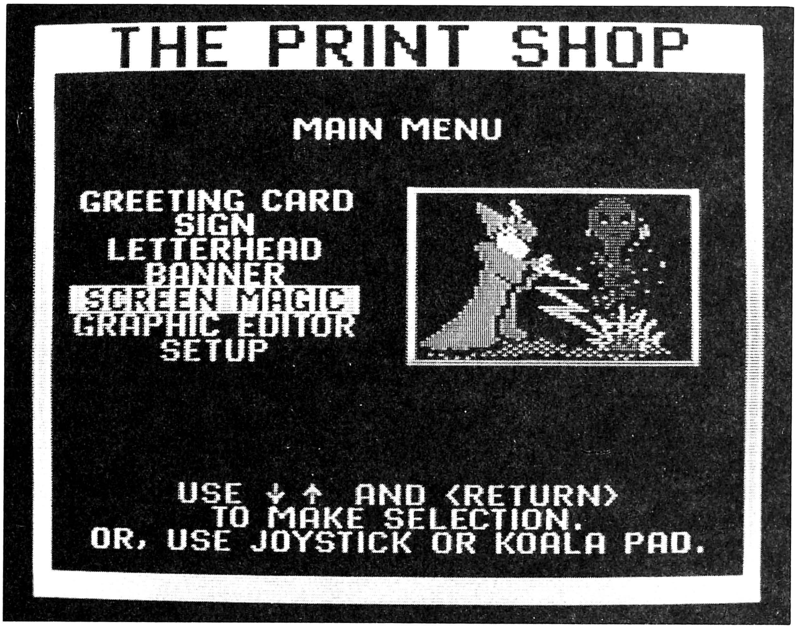
3. The "Get Screen" menu will tell you to enter the name of the screen or graphic you want to load and display, or press RETURN to see a list of graphics on the data disk. (See Figure 10.2.)

   We'll enter "Sevens," the name of a picture stored on my data disk. The picture will now appear on the screen.

4. Press RETURN again and the "Select Activity" menu will reappear. Choose "Draw Text on Screen."

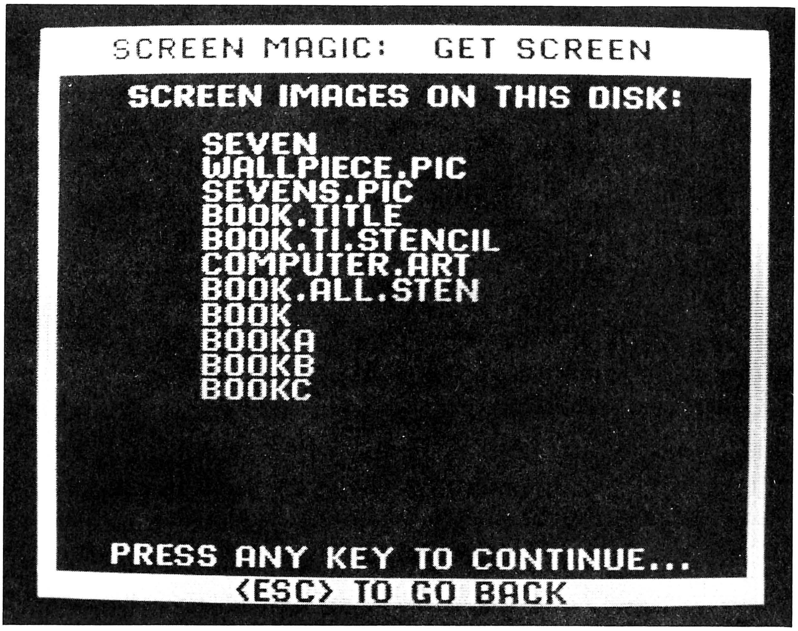
5. From the "Draw Text" menu choose "Draw Text on Screen in Memory." (This really means, "Draw Text on the Picture.")

6. The picture appears on the screen. Press RETURN.


THE PRINT SHOP
MAIN MENU
GREETING CARD
SIGN
LETTERHEAD
BANNER
SCREEN MAGIC
GRAPHIC EDITOR
SETUP

USE ↓ ↑ AND <RETURN> TO MAKE SELECTION. OR, USE JOYSTICK OR KOALA PAD.

Figure 10.1 *Print Shop Main Menu*



```
SCREEN MAGIC:   GET SCREEN
SCREEN IMAGES ON THIS DISK:
        SEVEN
        WALLPIECE.PIC
        SEVENS.PIC
        BOOK.TITLE
        BOOK.TI.STENCIL
        COMPUTER.ART
        BOOK.ALL.STEN
        BOOK
        BOOKA
        BOOKB
        BOOKC
PRESS ANY KEY TO CONTINUE...
      <ESC> TO GO BACK
```

Figure 10.2 *Print Shop Data Disk*

7. When the "Draw Text" menu reappears, select "Choose a Font." Press RETURN and an array of font samples will appear.
8. Choose "Stencil."
9. A new "Draw Text" menu will greet you with "Now type your message." A rectangular outline, sort of a "Compu-blackboard," appears on the screen, indicating that the stencil font will permit eight lines of text per screen. (Eighteen letters per line for the Stencil font, try it.)

   At the bottom of this "Draw Text" menu is the message "<CTRL–I> TO SEE EDIT INSTRUCTIONS." Use this if you need help.
10. Type "COMPUTER ART" on the top line. Press RETURN and type in "DESIGNED BY" on the next line. Press RETURN, and then, finally, enter your own name (instead of mine!) on the third line.

11. Press RETURN and the question "Center text top to bottom?" will appear at the bottom of the screen. Answer "Yes."

12. Now for some fun action! The picture reappears and your text will first be traced out in black and then filled in with white to create black-edged white letters on the picture.

Ta-Da! There's your book title with sparkling black-and-white letters on the colorful picture background, just the way you planned it! See Figure 10.3 (turn to the color illustrations following page 146).

Since the Print Shop's purpose is to produce printed material, let's go a few steps further and print out the book cover design. Remember, the picture is still on the screen.

13. Press RETURN and the "Select Activity" menu will appear.

14. Choose "Print." Press RETURN. The picture will return to the screen.

15. Press RETURN twice and a message at the bottom of the screen asks you to choose either NORMAL or REVERSED printing. Choose REVERSED (to make the text print in white letters).

16. Now indicate that you want to draw a frame around the picture. The "frame" is just a narrow line!

17. When the PRINT menu appears again, choose "TEST PAPER POSITION." Press RETURN and the printer will draw a dotted line that should fall on the perforation line between sheets of paper. If it doesn't, reposition the paper and rerun the test.

    Unless you want more than one copy of the printout, you can ignore the menu question regarding number of copies desired.

    Make sure the printer is ready to run.

18. Indicate that you want to print on the top half of the paper. Press RETURN, the printer will zip into action, and in a few moments you'll have your book cover design hot off the press!

More!

Ready to open another package? Chapter 11 will bring you the wonderful world of Beagle Graphics.

11

Beagle Graphics

What Is It?

The Beagle Graphics package, produced by Beagle Brothers Micro Software, is rated as one of the best for the Apple II series computers. It opens the door to double HiRes graphics. The package consists of a system disk and an excellent instruction manual. The double-sided disk provides both DOS 3.3 and ProDOS operating systems, so you can take your choice. The manual includes special instructions where necessary for one operating system or the other.

Graphics that you create with this package must be stored on a data disk as described in Chapter 7. Don't even think about trying to store a picture on the system disk, it's absolutely loaded!

In keeping with the Beagle Brothers' tradition, the Beagle Graphics disk is unlocked and unprotected. You are encouraged to make a backup disk for both operating systems. (But not for your friends!)

Hardware Requirements

To use Beagle Graphics, you will need an Apple IIe with 128K or an Apple IIc and a monitor that can handle the 80-column character size. In my opinion, computer graphics without color just shouldn't happen, so it's best to use a monochrome monitor for legibility of menus and other screen legends plus a good color monitor to display your graphics. If you have an RGB monitor this problem shouldn't exist.

Pointing or Graphics Input Devices

A number of graphics input devices (or double-plot pointing devices) can be used with Beagle Graphics. It will handle the Apple Mouse; Apple Graphics Tablet; Koala Pad with joystick, paddles, or a stylus; and, of course, cursor control via your computer keyboard.

Printing Your Graphics Creations

Unfortunately, the Beagle Graphics package has no provision for feeding your graphics data to a printer (known as "graphics dump" in the trade). No big problem. There is a Beagle Brothers package inelegantly named "Triple Dump" that will handle just about any computer/printer combination. Triple Dump makes it a snap to convert your creations into black-and-white prints.

Incidentally, Triple Dump has some clever features, including the ability to rotate your pictures, magnify either axis, and crop. It can be used to print out Hi- and LoRes pictures, 40- or 80-column text, and double Hi- or LoRes pictures. Triple Dump works with either DOS 3.3 or ProDOS operating systems.

Details You Wouldn't Want to Miss

The Beagle Brothers clan have cleverly combined two classic approaches to graphics in one package. One for the artist, the other for the programmer. Double high-resolution operation is possible in either case.

For the artist, Beagle Graphics provides a drawing board and a color painting system. The artist can operate in a free-hand mode or use features that automatically draw or paint perfect circles, squares, or rectangles. In addition, artists add text to a picture, or even transfer parts of one picture to another. These are just some of the functions that an artist can use without any knowledge of programming. (Show me an artist who wants to learn programming!)

Beagle Graphics gives graphics programmers new and powerful tools to extend their capabilities. For example: HiRes converter programs that make it possible to convert normal HiRes pictures to double HiRes; double LoRes graphics; utility programs to convert LoRes or HiRes programs to double LoRes or double HiRes; programming to create a "slide show"; a utility program to permit cutting an area out of a picture and pasting it back in the picture in a different location (or in another picture); a utility program called "Double.Scrunch" that can be used to compress picture data to reduce storage space requirements; and a companion program to unpack pictures that have been Double.Scrunched! (This is important because packed pictures must be unpacked before they can be loaded for display.)

Other features available to the programmer are: text output to the double HiRes screen (with several fonts), and a font editor to permit alteration of letter styles.

The Menus

The Beagle Graphics system is menu-driven. When you start up the system disk, you are presented with a list of features or functions identified by single letters. Entering the letter signifying your choice on this and all subsequent menus gets the action underway. All entries must be made from the keyboard.

A Solid Gold Paperback?

Well, not quite. However, the Beagle Graphics instruction manual is a treasure chest of information. The first third of the manual deals with the drawing program for artists, designers, and others without interest in programming.

The remainder of the manual describes the function and use of every command related to double HiRes graphics and how to use them in programs.

Written in a chatty, confidence-inspiring style, this manual would be hard to beat with respect to clarity and organization. Unlike most software manuals, it has a complete index—a big plus when you're trying to learn a new system. In addition, the manual contains helpful program examples for the majority of the nearly three dozen double HiRes commands. A spiral binding makes the manual easy to use at the side of your keyboard. I rate this manual among the very best for software packages.

After this enthusiastic description of Beagle Graphics, you're no doubt ready for a demonstration. The following pages will show you how easily a typical project is handled by Beagle Graphics' nonprogramming graphics feature called Double Plot.

Beagle Graphics at Work

The next few pages will walk you through the steps an artist would take to create a poster or fabric design using the Beagle Graphics Double Plot program.

Create a Poster or Fabric Design

As mentioned in previous chapters, there's just no substitute for hands-on experience. However, following the steps necessary to create a design on the computer screen will help you visualize how you can use this software package to enhance your own creative talents.

The approach to this project will be: to display a previously stored picture, to select and store a small area of it, to remove the displayed picture from the screen, to convert the screen to a blank white area, and finally, to print repeatedly the stored area of the displayed picture on the screen to create a pleasing design.

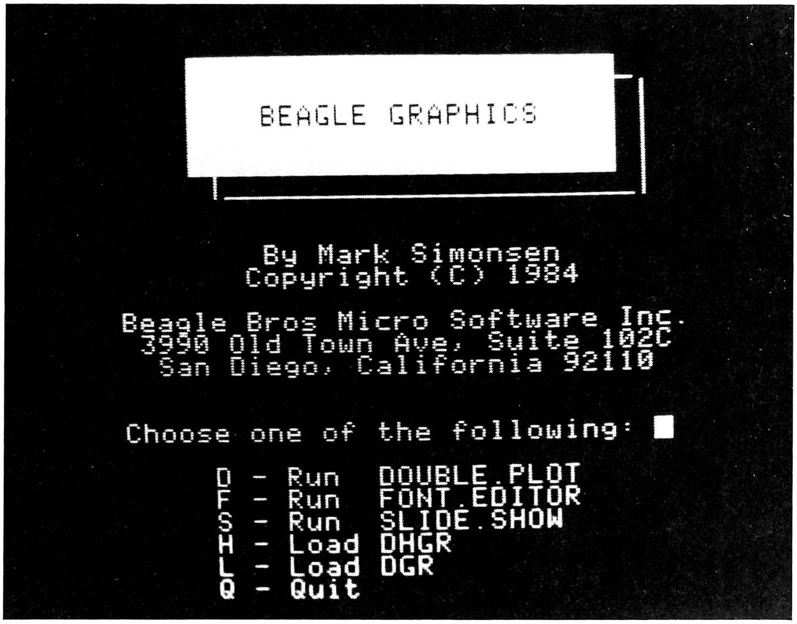
What's Needed for this Project?

For this project, Beagle Graphics assumes that you have an Apple IIe with an 80-column board or an Apple IIc. A two-disk system is desirable but not a must. A graphics input device is needed. Let's use a mouse. And, of course, for storing any designs, a data disk.

The Beagle Graphics Double Plot program requires a monitor capable of legibly displaying Apple 80-column text. For this program, I use two monitors: an Apple monochromatic monitor (for legibility of text) and a good composite video color monitor to display the graphics. As mentioned earlier, an RGB monitor should handle both requirements very well.

Getting Started

Beagle Graphics operates in either DOS 3.3 or ProDOS. For our projects, let's choose the ProDOS side of the system disk—and make sure that any pictures we might want to load are in ProDOS, too!



```
BEAGLE GRAPHICS

By Mark Simonsen
Copyright (C) 1984

Beagle Bros Micro Software Inc.
3990 Old Town Ave, Suite 102C
San Diego, California 92110

Choose one of the following: ■

D - Run  DOUBLE.PLOT
F - Run  FONT.EDITOR
S - Run  SLIDE.SHOW
H - Load DHGR
L - Load DGR
Q - Quit
```

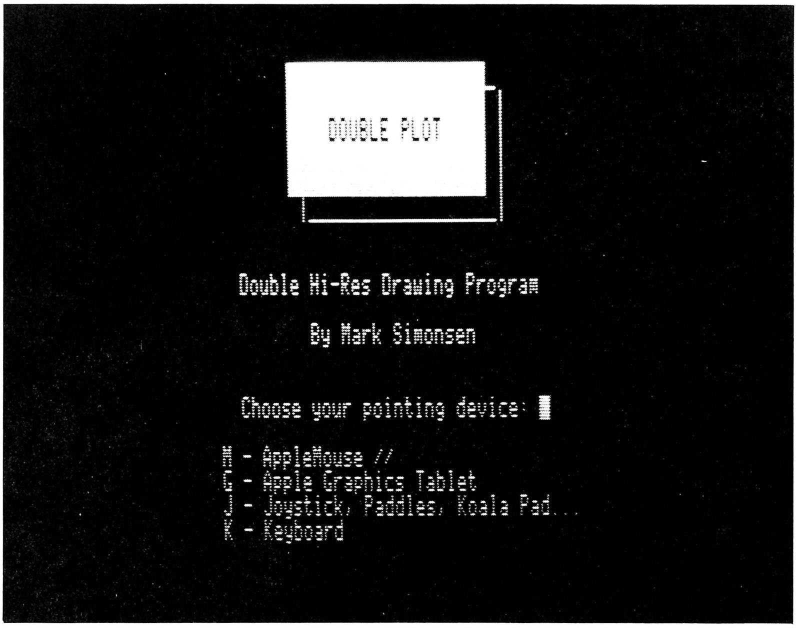
Figure 11.1 ***Beagle Graphics Main Menu***

Now it's time to boot up the system disk. After the startup operation is complete, the Beagle Graphics opening menu greets you on the screen as shown in Figure 11.1.

1. Choose option D to run the Double Plot program.
2. After the Double Plot program has loaded, a pointing device menu appears, requesting a choice among pointing devices. See Figure 11.2. Enter M, for mouse.
3. Now the main Double Plot menu will appear across the bottom of the working area. This menu gives you access to any of the twelve operations listed by entering just one letter. Now that's the way to make users friendly!

At this point we need to LOAD a picture (named "Triangles") so we can snip out a very small area to be used to create a poster or fabric design by repeated "printing" on the blank white screen. The cut-and-paste feature of the Double Plot program makes it easy to perform this kind of magic. But first, let's get "Triangles" up on the screen. This picture was previously stored on a data disk.

4. Enter CRTL D and the Disk menu will appear.
5. Remove the system disk and replace it with the data disk containing "Triangles."
6. Choose the Load option by entering L.
7. Respond to the NAME OF PICTURE? question with TRIANGLES.
8. After the picture appears on the screen, the main menu will reappear at the bottom of the screen. Choose the Edit option and enter E.



```
DOUBLE PLOT

Double Hi-Res Drawing Program

By Mark Simonsen

Choose your pointing device:

M - AppleMouse //
G - Apple Graphics Tablet
J - Joystick, Paddles, Koala Pad...
K - Keyboard
```

Figure 11.2 *Beagle Pointing Device Menu*

9. Now the image of a pair of scissors will appear somewhere on the "Triangles" picture. To cut out an area of the picture (to be used in making the fabric design), use the mouse to move the scissors to the upper left corner of the area you want.
10. Hold the mouse button down and use the mouse to move the scissors diagonally to the lower right until the "elastic box" connected to the scissors contains a desired area. Figure 11.3 shows symbolically how this elastic box appears on the screen.
11. Now enter M, and the selected picture area will be stored for later use.
12. To get a clean screen for our fabric design, we'll enter X.
13. Answer the CLEAR THE SCREEN TO WHAT COLOR question with P, indicating that we want a white screen.

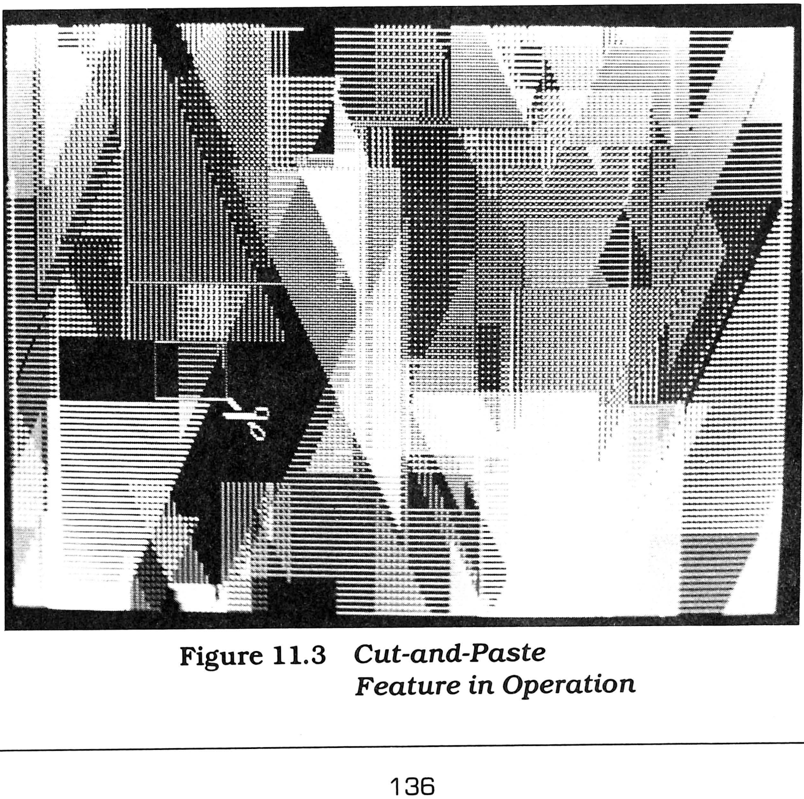


Figure 11.3 *Cut-and-Paste Feature in Operation*

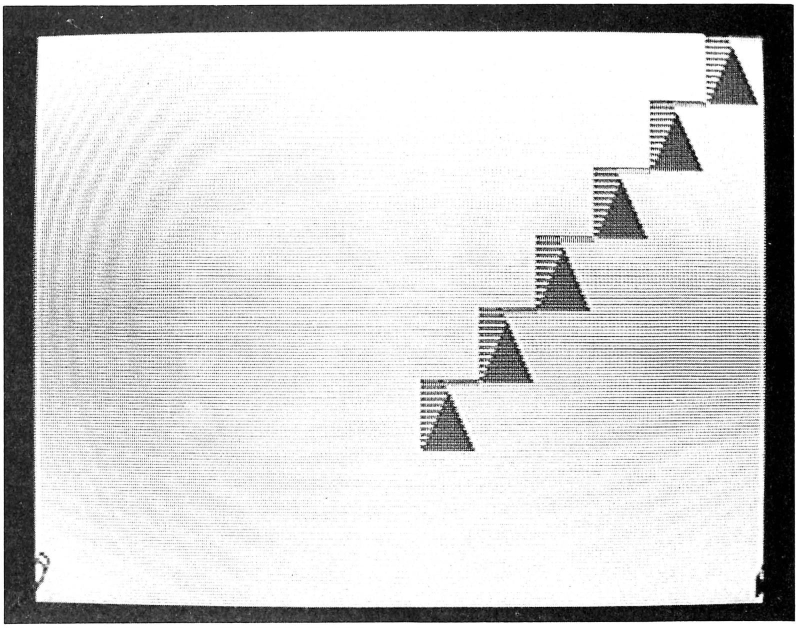


Figure 11.4 *Initial Steps of Fabric Design*

14. With the main menu showing at the bottom of the screen, press Left-Apple and E (both keys at the same time). This will activate the paste mode and you can use the mouse to place the stored area anywhere on the screen.

All of the steps up to this point have been preparatory to actually creating a design. Now's a good time to make some pencil sketches of some design possibilities and choose one that looks achievable on the screen. You can also experiment right on the screen using the "X" command to clear the screen as you try out various design ideas.

Figure 11.4 shows the initial steps of the final design of Figure 11.5. Each step in the composition of your design is carried out by your careful manipulation of the mouse. Figure 11.6 (see the color illustrations following page 146) demonstrates a design created by using other areas of "Triangles."

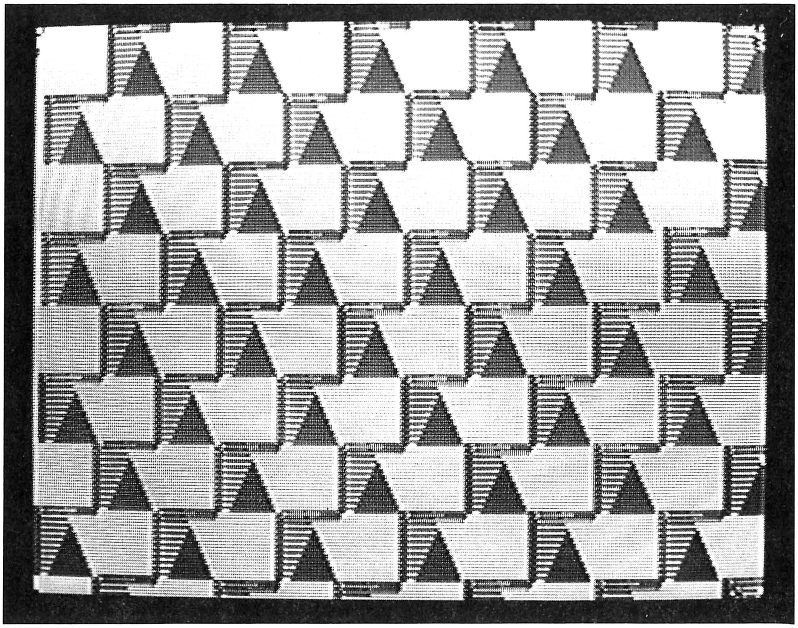


Figure 11.5 ***Final Fabric Design***

The Finale

The designs shown here demonstrate how the Double Plot programs can be used for creative expression purposes. There are many additional features that could be put to work to expand the design possibilities of the program. A few "for instances": draw a cartoon character, then cut and paste it randomly on a colored background for use as a textile pattern or wrapping paper design, use single letters of text or words as a pattern. Why not? Put your imagination to work!

By the way, if your imagination says, "Dazzle Draw software won't display Beagle Graphics pictures, so what if I want to use Dazzle Draw's ZOOM feature to enhance my nifty Beagle Graphics design?" No problem! Chapter 12 will show you how to fit your Beagle Graphics pictures into Dazzle Draw's software (and vice versa). A zoom feature is also included in a recent update of Beagle Graphics.

12

Conversion and Other Conveniences

Yin and Yang

There are a number of reasons why you might wish to display or modify Dazzle Draw pictures while using the Beagle Graphics Double Plot system, or vice versa. For example, you might have a number of Dazzle Draw pictures that you would like to display with a series of Double Plot pictures—or you might wish to modify a Double Plot picture using a feature available only with the Dazzle Draw software package. This kind of manipulation is possible only if you convert each picture file to the correct format for the graphics package involved. This sounds like a rather formidable task, but it can be done easily using conversion programs listed in this chapter and included on the disk available with this book. Here's a little background information on the subject.

About Picture Files

Let's consider a few basic facts about how *picture files* are handled. Dazzle Draw, for instance, saves each picture to one file with one name, such as PICTURE. Beagle Graphics saves pictures to two files, which would appear on the disk catalog as

PICTURE and PICTURE.AUX. For the Complete Graphics System, it's PICTURE.DPM and PICTURE.DPA, respectively. More on this, including an example, in Chapter 13.

Speaking of standards or conventions, graphics packages do not all function in the same operating system. Dazzle Draw operates in the ProDOS system, Complete Graphics System in DOS 3.3, and Beagle Graphics in either.

Sounds Complicated? It Isn't!

If you are thinking, "This whole business of converting pictures sounds pretty complicated"—don't worry! You have all the tools you need, and the programs that follow in this chapter will make it easy. (By the way, now's a good time to mention that Programs 12.1, 12.2, and 12.3 operate in ProDOS.)

As an example of picture conversion, let's say that you want to convert the Beagle Graphics DOS 3.3 file PICTURE for use with the Dazzle Draw package. (Remember, the conversion program works in ProDOS, so you must first get the picture data from DOS 3.3 in ProDOS.)

1. Convert the DOS 3.3 file PICTURE to ProDOS, using either the ProDOS user's disk or the ProDOS system utility disk.
2. Use Program 12.1 to convert the file from Beagle Graphics to Dazzle Draw.

(If the PICTURE file had been a ProDOS file to begin with, you could have gone directly to the software conversion Program 12.1.)

Beagle to Dazzle and Vice Versa

Let's assume that you want to touch up or modify a Beagle Graphics picture (in ProDOS format), using some of the Dazzle

Draw system features. Remember, before you can display your Beagle Graphics picture in the Dazzle Draw system, you need to give it a single filename. Program 12.1 will do this name conversion for you in seconds.

Load the program and when it requests "Enter Pathname With Filename—BEAGLE GRAPHICS FILENAME," remove the program disk, and insert the picture (data) disk. *Then* enter the Beagle Graphics picture name. Be sure to assign a different name for the Dazzle Draw filename (or use the same name with DD added). Also be certain that the disk that will receive the converted picture has sufficient storage space. Do *not* use quotation marks around picture names in software conversion programs.

First, Beagle to Dazzle

Program 12.1 **Convert Beagle Graphics to Dazzle Draw**

```
 5 REM PROGRAM 12.1 CONVERT BEAGLE GRAPHICS TO
   DAZZLE DRAW
 10 REM WRITTEN BY R.R. PRESCOTT
 20 DIM B$ (60), C$(60), D$(60)
 30 HOME
 40 LOMEM:24576
 50 ONERR GOTO 150
 60 PRINT "CONVERT BEAGLE GRAPHICS PICTURES"
 70 PRINT "TO DAZZLE DRAW PICTURES"
 80 PRINT "ENTER PATHNAME WITH FILENAME"
 90 PRINT "UNLESS DEFAULT PATHNAME"
 100 "INPUT BEAGLE GRAPHICS FILENAME"; B$
 110 C$ = B$+ ".AUX"
 120 PRINT CHR$(4)"BLOAD" C$; ",A$2000"
 130 PRINT CHR$(4)"BLOAD" B$; ",A$4000"
 140 GOTO 190
 150 PRINT CHR$(7)"INVALID BEAGLE GRAPHICS
     PATHNAME!!"
 160 FOR I = 1 TO 1000: J = J: NEXT
```

(continued on the next page)

```
170  GOTO 30
180  ONERR GOTO 260
190  INPUT "DAZZLE DRAW FILENAME"; D$
200  PRINT CHR$(4) "BSAVE" D$; ",A$2000, L$3FF8"
210  PRINT "DONE!!"
220  FOR I = 1 TO 500 : J = J : NEXT
230  INPUT "DO ANOTHER CONVERSION (Y/N) " ; E$
240  IF E$ = "y" OR E$ = "Y" THEN 30
250  END
260  PRINT CHR$(7) "INVALID DAZZLE DRAW PATHNAME!!"
270  FOR I = 1 TO 1000 : J = J : NEXT
280  GOTO 30
```

Dazzle to Beagle, Over!

Now, if your stock of stored pictures is strictly from Dazzle Draw, you'll need Program 12.2 to convert them to good old Beagle Graphics!

Program 12.2 **Convert Dazzle Draw to Beagle Graphics**

```
5  REM PROGRAM 12.2 CONVERT DAZZLE DRAW TO BEAGLE
   GRAPHICS
10  REM WRITTEN BY R.R. PRESCOTT
20  DIM B$(60), C$(60), D$(60)
30  HOME
40  ONERR GOTO 120
50  PRINT "CONVERT DAZZLE DRAW PICTURES"
60  PRINT "TO BEAGLE GRAPHICS PICTURES"
70  PRINT "ENTER PATHNAME WITH FILENAME"
80  PRINT "UNLESS DEFAULT PATHNAME"
90  INPUT "DAZZLE DRAW FILENAME"; D$
100  PRINT CHR$(4) "BLOAD" D$ ",A$2000"
110  GOTO 150
120  PRINT CHR$(7) "INVALID DAZZLE DRAW FILENAME!!"
130  FOR I = 1 TO 1000 : J = J : NEXT
140  GOTO 30
```

```
150 ONERR GOTO 240
160 INPUT "BEAGLE GRAPHICS FILENAME"; B$
170 C$ = B$ + ".AUX"
180 PRINT CHR$(4) "BSAVE" B$ ",A$4000, L$2000"
190 PRINT CHR$(4) "BSAVE" C$ ",A$2000, L$2000"
200 PRINT "DONE!!"
210 INPUT "DO ANOTHER CONVERSION (Y/N)"; E$
220 IF E$ = "y" OR E$ = "Y" THEN 30
230 END
240 PRINT CHR$(7) "INVALID BEAGLE GRAPHICS
    PATHNAME!!"
250 FOR I = 1 TO 1000 : J = J : NEXT
260 GOTO 30
```

For Complete Graphics System Fans Only

Here's an unadvertised "special." There's nothing like converting a conversion program! Change a half a dozen lines of Program 12.1 or Program 12.2 and you get Program 12.3, which enables you to convert Dazzle Draw pictures for display via The Complete Graphics System.

Program 12.3 **Convert Dazzle Draw to Complete Graphics System**

```
5 REM PROGRAM 12.3 CONVERT DAZZLE DRAW TO COMP.
  GR. SYS.
10 REM WRITTEN BY R.R. PRESCOTT
20 DIM B$(60), C$(60), D$(60)
30 HOME
40 ONERR GOTO 120
50 PRINT "CONVERT DAZZLE DRAW PICTURES"
60 PRINT "TO COMP. GR. SYS. PICTURES"
70 PRINT "ENTER PATHNAME WITH FILENAME"
80 PRINT "UNLESS DEFAULT PATHNAME"
90 INPUT "DAZZLE DRAW FILENAME"; D$
```

(continued on the next page)

```
100 PRINT "CHR$(4)"BLOAD" D$ ",A$2000"
110 GOTO 150
120 PRINT CHR$(7)"INVALID DAZZLE DRAW FILENAME!!"
130 FOR I = 1 TO 1000 : J = J : NEXT
140 GOTO 30
150 ONERR GOTO 250
160 INPUT "COMP. GRAPHICS SYSTEM FILENAME"; B$
170 C$ = B$ + ".DPA"
180 D$ = B$ + ".DPM"
190 PRINT CHR$(4)"BSAVE" D$ ",A$4000,L$2000"
200 PRINT CHR$(4)"BSAVE" C$ ",A$2000,L$2000"
210 PRINT "DONE!!"
220 INPUT "DO ANOTHER CONVERSION(Y/N)?"; E$
230 IF E$ = "y" OR E$ = "Y" THEN 30
240 END
250 PRINT CHR$(7)"INVALID COMP. GRAPHICS SYS.
    PATHNAME!"
260 FOR I = 1 TO 1000 : J = J : NEXT
270 GOTO 30
```

Converting HiRes Pictures to Double HiRes

How about converting some of your high-resolution graphics pictures to double HiRes? The Beagle Graphics instruction manual tells you how to perform this magic easily from your keyboard, but if you have many pictures to convert, using a program to do the job will speed things up considerably.

After you convert your pictures to the double HiRes form, you'll surely want to save them.

Here's a program that does both jobs with the greatest of ease. It will convert your pictures and save them—and all you have to do is enter picture names (use capitals and don't forget the "quotes" before and after names!).

Program 12.4 **Convert HiRes Pictures to Double HiRes and Save**

```
5 REM PROGRAM 12.4 CONVERT HIRES TO DOUBLE HIRES
  AND SAVE
10 REM DERIVED FROM BEAGLE GRAPHICS INST. MANUAL
   DATA
20 PRINT "ENTER NAME OF PICTURE TO BE CONVERTED TO
   DOUBLE HIRES IN QUOTES."
30 INPUT B$
40 PRINT CHR$(4) "BRUN DHGR"
50 PRINT CHR$(4)"BLOAD CONVERT. HIRES.2"
60 & HGR:VTAB 21
70 PRINT CHR$(4);"BLOAD";B$;",A$4000"
80 CALL 768
90 POKE 49234,0:FOR N=1 TO 4000:NEXT N
100 POKE 49235,0
110 VTAB 21:PRINT "IF YOU WANT TO SAVE THIS DOUBLE
    HIRES PICTURE, ENTER NEW NAME IN QUOTES"
120 INPUT A$
130 B$=A$+".AUX"
140 & SAVE A$,B$
150 POKE 49234,0
NOTE: BE SURE TO INCLUDE THE , BEFORE A$4000 IN LINE
      70 AND THE . BEFORE AUX IN LINE 130.
```

Saving Double HiRes Pictures

Now, here's the SAVE part of Program 12.4 for use with pictures already in the double HiRes form.

Program 12.5 **Save DHR Pictures**

```
10 REM PROGRAM 12.5 SAVE DHR PICTURES
20 PRINT "ENTER NAME OF PICTURE IN QUOTES"
30 INPUT A$:B$=A$+".AUX"
40 & SAVE A$,B$
```

Displaying Double HiRes Pictures

Another program that will save you time and typing is this one for showing the double HiRes pictures you made using the Beagle Graphics system, or converted to that system. Keep this program in mind for displaying pictures created by the programming methods described in Chapter 13.

Program 12.6 **Display DHR Pictures**

```
5  REM PROGRAM 12.6 DISPLAY DHR PICTURES
10 PRINT CHR$(4)"BRUN DHGR"
20 PRINT "ENTER NAME OF PICTURE TO BE DISPLAYED"
30 INPUT A$ : B$=A$+".AUX"
40 & HGR2 : & LOAD A$,B$
```

The End Is Near

The next, and almost final, chapter of this book returns to generation of images by programming, but now with the higher levels of sophistication incorporated in Beagle Graphics double HiRes programming commands. The next chapter also tells you how to convert programs from HGR to DHGR.

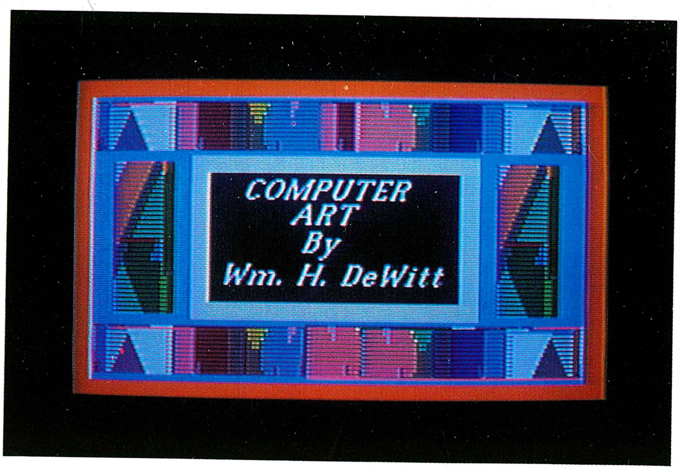

COMPUTER
ART
By
Wm. H. DeWitt

Figure 8.5
Dazzle Draw
Cut-and-Paste Design

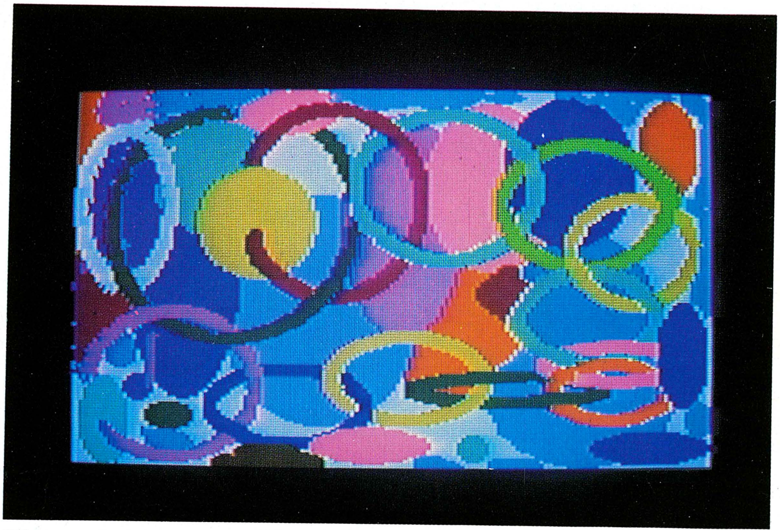


Figure 8.6 *A Solar Sorority*

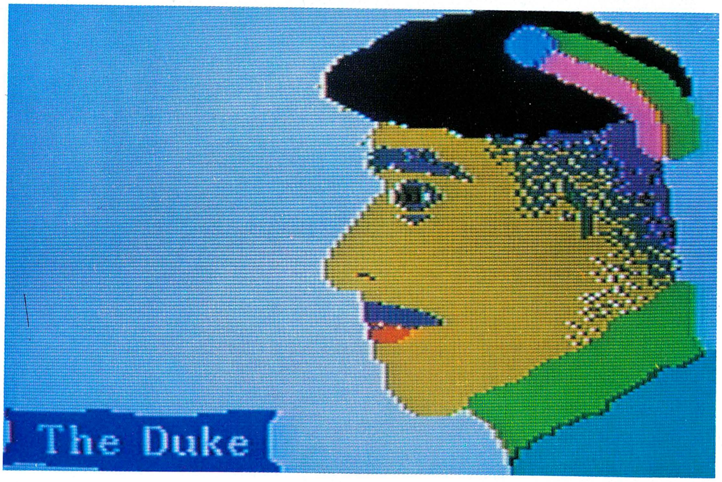


Figure 8.7 *The Duke*

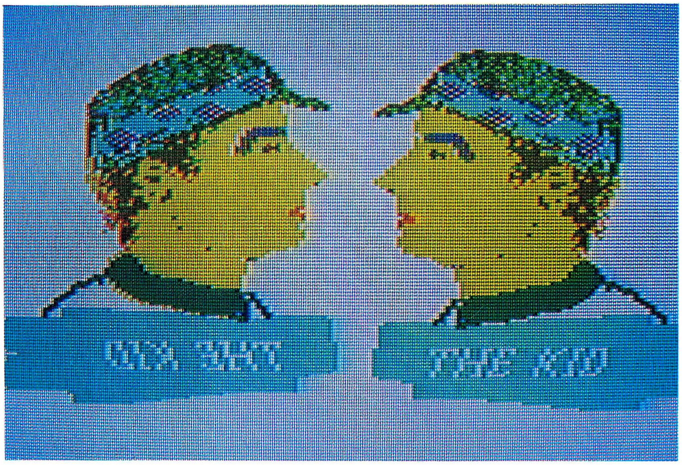


Figure 8.8 *The Kid*

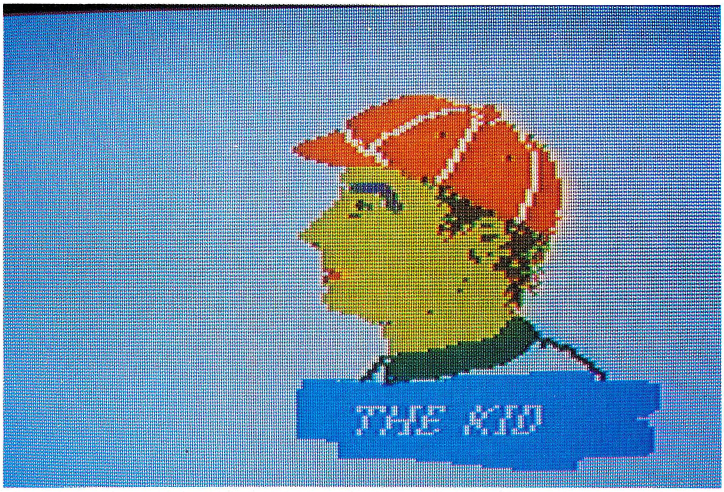


Figure 8.9 *The Kid in a Red Cap*

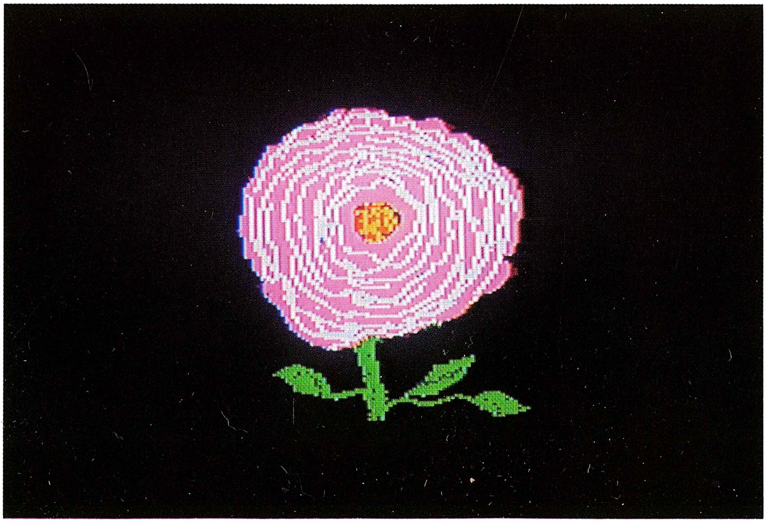


Figure 8.10 *La Rosa*

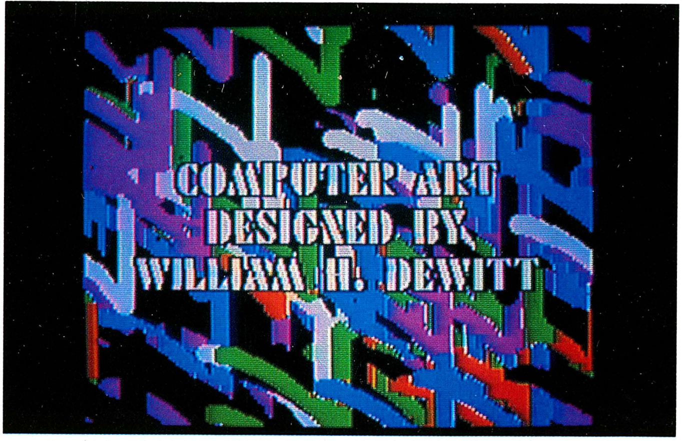

COMPUTER ART
DESIGNED BY
WILLIAM H. DEWITT

Figure 10.3
Book Cover Computer Art

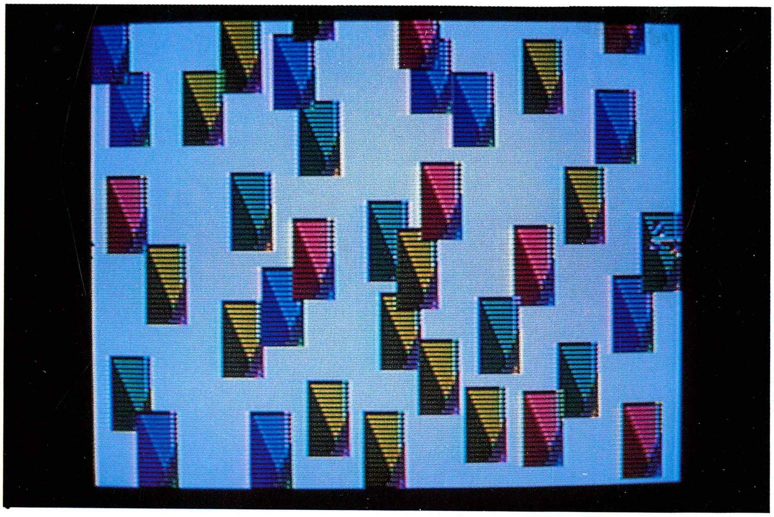


Figure 11.6 *Second Fabric Design*

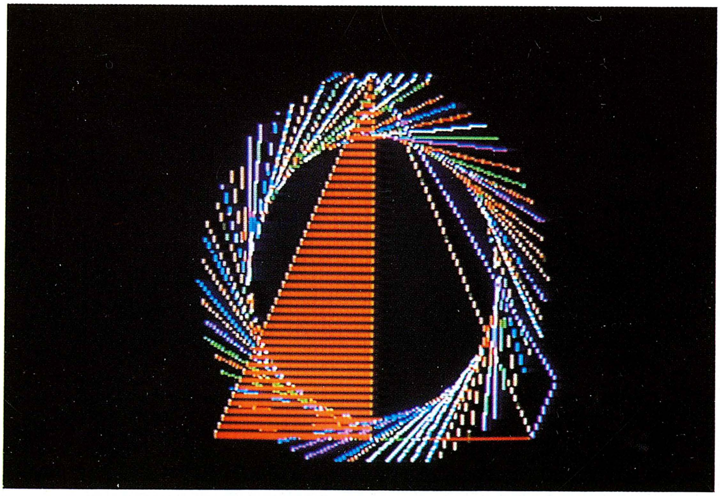


Figure 13.1 *Shape Rotation—HGR Mode*

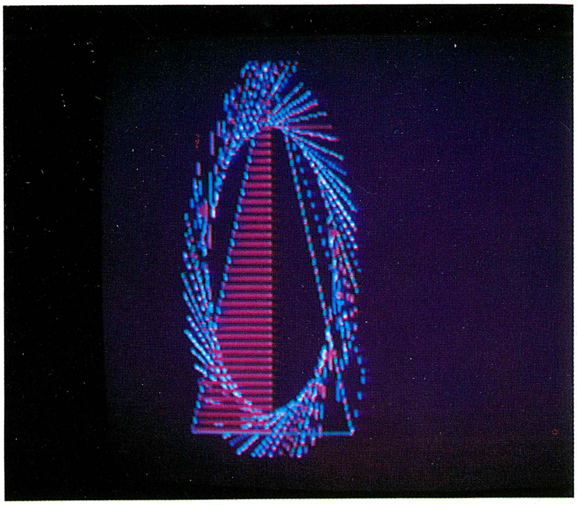


Figure 13.2 *Same Shape and Rotation—DHGR Mode*

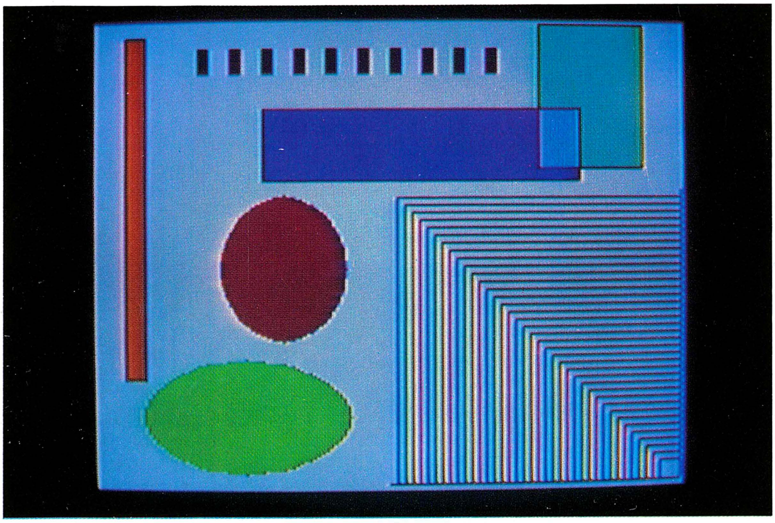


Figure 13.3 *Ampersand/DHGR Command Demonstration*

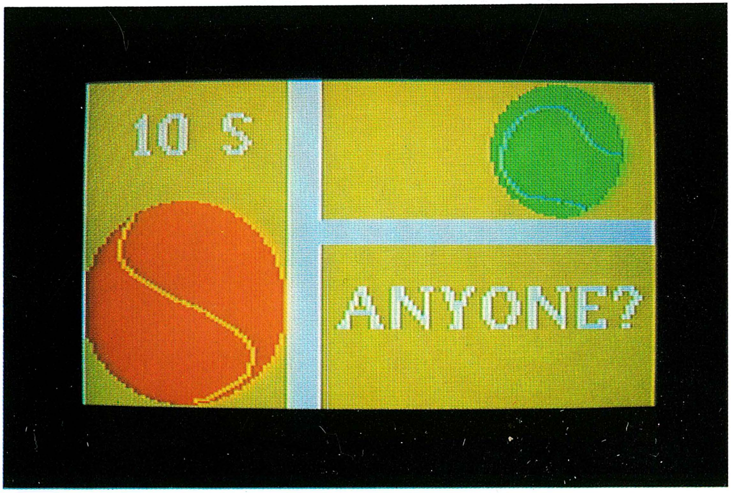


Figure 13.4 *Tennis Anyone? Demo Slide*

13

Programming in Double HiRes

Doubles, Anyone?

Beagle Graphics software is a delight to users who want to do their own programming in the double HiRes mode, because the software is complete and the manual is coherent.

This chapter gives you programmers a sample of Beagle Graphics instructions and programs to extend your double HiRes graphics capabilities. However, if you buy Beagle Graphics, you'll quickly recognize that what you see in this chapter is only a small sample of the features jampacked into this nifty package.

An Acknowledgment

All program material in this chapter is based on the use of the Beagle Graphics system disk and information from the instruction manual.

Ampersand commands mentioned in this chapter are explained in relation to working programs. For a complete listing and explanation of all Ampersand commands, see the Beagle Graphics instruction manual.

Three Important Reminders

1. The Beagle Graphics software is contained in both DOS 3.3 and ProDOS operating systems on a two-sided disk. If you don't want any "back talk" from your Apple, be sure to choose the format that matches your data disk (or vice versa).
2. If you want to use "old" DOS 3.3 pictures or programs in a ProDOS system, you must convert them to double HiRes first and then use the ProDOS system disk to convert them to ProDOS.
3. You must BRUN the DHGR utility to enable the double HiRes (Ampersand) commands.

A Timing-Saving Suggestion

If you plan to do very much double HiRes programming, it's worthwhile to make up a special data disk—call it "/PROGRAMS"—that has startup capability plus a few utility programs on it. In this way you can avoid going back and forth between your data disk and the system disk. By including the ProDOS and Basic.System routines, you can make other uses of the disk if you wish.

For your "/PROGRAMS" disk, copy the files listed on the next page from the Beagle Graphics system disk.

Table 13.1

File	Blocks Used
1. PRODOS	31
2. BASIC.SYSTEMS	21
3. DHGR	10
4. STARTUP PROGRAM	1
5. HGR.TO.DHGR	1
6. CONVERT.HIRES.2	1
TOTAL BLOCKS USED	65 (Or about 25% of disk)

Startup

In the ProDOS system, the word STARTUP is recognized by the system and causes the loading of the ProDOS and Basic.System files. A startup program is much the same as the HELLO program in DOS 3.3.

Here is a very short startup program that will let you know what's on the disk and set you up for graphics programming. Save this program with the filename STARTUP for use on your data disks.

Program 13.1 **Startup**

```
5 REM PROGRAM 13.1 STARTUP
10 PRINT CHR$(4)"BRUN DHGR"
20 PRINT CHR$(4)"PR#3"
30 PRINT CHR$(4)"CATALOG"
```

Easy Does It

The easiest way to write a double HiRes program is to convert one of your favorite normal HiRes programs to double HiRes using the Beagle Graphics routine called HGR.TO.DHGR.

The instruction manual outlines the following three-step procedure using a utility program included on the system disk. Here's how it works—*after* you have BRUN DHGR:

1. Load the HGR.TO.DHGR utility:

   ```
   BLOAD HGR.TO.DHGR
   ```

2. Load the Applesoft program to be converted:

   ```
   LOAD program
   ```

3. Call the Ampersand inserter routine:

   ```
   CALL 768
   ```

The machine language program of step 1 enables the computer to operate with the Ampersand commands. Step 2 puts the normal HiRes program in memory. Step 3 inserts Ampersands in front of all normal HiRes commands in the program to convert them to double HiRes.

Does it work? We'll try it and see.

A Program Conversion, HGR to DHGR

Let's take a short HGR program called "Pulsar" that creates a multicolored pulsar-like image and convert it to the DHGR form. After it's converted, we'll see what else must be done to make full use of the DHGR features. Here's how to do it:

1. Assuming that you have the Beagle Graphics system disk in Drive 1, enter BLOAD HGR.TO.DHGR, and then BRUN DHGR. (Or use a data disk as described above.)

2. Remove the system disk, insert a data disk with the "Pulsar" program on it.
3. Enter LOAD PROGRAM 13.2 PULSAR
4. CALL 768
5. SAVE the converted program to the data disk. Call the program "Program 13.3 DHR.PULSAR."

Now Compare the Old and the New

Here's the old HGR program.

Program 13.2 **Pulsar**

```
1 REM PROGRAM 13.2 PULSAR
10 HGR : POKE 49234,0
20 HCOLOR= INT(RND(1)*8)
25 REM PUT STARS IN THE SKY
30 HPLOT INT(RND(1)*279), INT(RND(1)*190)
40 X= INT(RND(1)*279)
50 Y= INT(RND(1)*190)
60 HPLOT X,Y
65 REM MAKE A PULSAR
70 HPLOT TO 140,90
80 GOTO 20
```

After "Ampersanding"

Here's the DHGR version.

Program 13.3 **DHR.Pulsar**

```
1  REM PROGRAM 13.3 DHR.PULSAR
10  & HGR : POKE 49234,0
20  & HCOLOR= INT(RND(1)*8)
25  REM PUT STARS IN THE SKY
30  & HPLOT INT(RND(1)*279), INT(RND(1)*190)
40  X= INT(RND(1)*279)
50  Y= INT(RND(1)*190)
60  & HPLOT TO 140, 90
65  REM MAKE A PULSAR
70  & HPLOT TO 140,90
80  GOTO 20
```

Now every HiRes command is preceded by an Ampersand, but further changes are needed to access the 16 colors of DHGR, make use of the 560 horizontal points across the screen, and to center the principal image on the screen. Here are the changes:

1. Line 1: Change name to Program 13.4 Final DHR.PULSAR
2. Line 5: Add BRUN DHGR to enable Ampersand commands.
3. Line 10: Change & HGR to & HGR2 to eliminate need for POKE 49234,0.
4. Line 20: Change 8 to 16 to increase color gamut.
5. Line 30: Change 279 to 560 to make use of full screen width for star positions.
6. Line 40: Change 279 to 560 to make use of full screen width for "pulsar" radii.

7. Line 70: Change 140 to 280 to center the star on the screen.

Note: Copy the DHGR utility from the Beagle Graphics system disk onto your data disk using the ProDOS system disk. To avoid "disk swapping," you may wish to copy other utilities at the same time.

Custom Finish

And now, here's the DHGR version of the original program, with adjustments made to increase the color gamut to 16 colors and to make use of the full 560 pixel screen width:

Program 13.4 **Final DHR.Pulsar**

```
1 REM PROGRAM 13.4 FINAL DHR.PULSAR
5 PRINT CHR$(4)"BRUN DHGR"
10 & HGR2 :REM ELIM. POKE 49234,0
20 & HCOLOR= INT(RND(1)*16):REM CHANGE 8 TO 16
25 REM PUT STARS IN SKY
30 & HPLOT INT(RND(1)*560), INT(RND(1)*190):REM
   CHANGE 279 TO 560
40 X= INT(RND(1)*560):REM CHANGE 279 TO 560
50 Y= INT(RND(1)*190)
60 & HPLOT X,Y
65 REM MAKE A PULSAR
70 & HPLOT TO 280,90:REM CHANGE 140 TO 280
80 GOTO 20
```

Now that all changes have been made, why not SAVE this short DHGR program for future reference?

All That Glitters

The nearly perfect wreathlike circle of Figure 13.1 (see color illustrations following page 146) was created by rotating a shape in the regular HiRes (HGR) mode. When the same shape data is used in the double HiRes mode, the rotated shape produces the ellipsoidal form shown in Figure 13.2 (see color illustrations following page 146). This problem is caused by the difference in the X/Y aspect ratio between HGR and DHGR. Nothing can be done about it in Applesoft BASIC programming. You can partially circumvent this problem using the following technique:

1. Use programming in HGR to create the parts of the desired image that call for circular patterns.
2. Store those images as a picture.
3. Convert the picture to double HiRes using Beagle Graphics' CONVERT.HIRES.2 PICTURE conversion routine and & SAVE it. See pages 55 and 83 of the Beagle Graphics instruction manual. (Be sure to give the picture a new DHGR-indicating name.)
4. & LOAD the converted picture as part of a DHGR program.
5. Add other images to the picture as desired by programming with DHGR facilities.

In this rather roundabout way you can capture the advantages of DHGR for all but circular shape-formed images.

Programming in DHGR from Scratch

First, a few minor explanations: The command & BOX(X length, Y length) AT (X,Y) draws a square or rectangle. If only one value is used with the & BOX part of the command, it will be used for both X and Y lengths. The AT X,Y portion of the command specifies the location of the upper left corner of the box.

The same rules apply to the commands & XBOX, & CIRCLE, and & XCIRCLE.

With the command & FILL(1st color,2nd color) AT X,Y, the double HiRes color of the FILL is specified by one or two numbers within the parentheses. The FILL must be located within an area that has a border. The black borders of the circles can be removed *after* the & FILL by using the & XCIRCLE command. (See next program.)

The command "& BCOLOR= color value" is followed by & CLEAR to clear the screen to the chosen color. The sequence of these commands is important.

There are four double HiRes graphics modes. The command & MODE (mode number) is used to select among:

1. 560 mode
2. 140 mode
3. 560 mixed mode
4. 140 mixed mode

See the instruction manual for further details.

Double HiRes from the Start

Here's a program that illustrates the performance of some frequently used Ampersand commands.

Figure 13.3 (see color illustrations following page 146) shows what happens at the end of the next 28 lines, but you can get some personal experience with the Ampersand command family by making changes and additions to this program. Try some variations, draw a design on paper and see how closely you can replicate it, change colors, add a border, see what happens when you use & FILL in the *wrong* place—experimenting is fun!

If you develop a design that you like, as I did in Figure 13.4 (see color illustrations), & SAVE it immediately instead of wishing you had done so. As always, hands-on experience is the best teacher.

Program 13.5 **Ampersand Demo**

```
5 REM PROGRAM 13.5 AMPERSAND DEMO
10 REM FROM BEAGLE GRAPHICS DATA
20 PRINT CHR$(4)"BRUN DHGR"
30 & HGR2 : & MODE(1) : & BCOLOR=15 : & CLEAR
40 & HCOLOR=0 : & BOX(100,60) AT 40,130
50 & BOX(20,140) AT 510,45: & FILL(9) AT 515,135
60 & BOX(300,30) AT 100,125: & FILL(2) AT 200,150
70 & FILL(14) AT 50,170: & FILL(7) AT 110,150
80 FOR X= 180 TO 460 STEP 30: Y= 170
90 & BOX(10) AT X,Y: NEXT X
100 FOR D= 185 TO 455 STEP 30
110 & FILL (0) AT D,175: NEXT D
120 & HCOLOR=0: & CIRCLE(60,30) AT 380,90
130 & FILL(1) AT 380,90: & XCIRCLE(60,30) AT 380,90
140 & CIRCLE(100,25) AT 415,30: & FILL(12) AT 415,30
150 & XCIRCLE(100,25) AT 415,30: & HCOLOR=0
160 A= 1
170 X= 20: Y= 8
180 IF X>280 THEN GOTO 230
190 & BOX(X,Y) AT 2,2
200 X= X+7: Y= Y +3
210 A= A+1: IF A= 114 THEN END
220 GOTO 180
230 IF X<0 THEN GOTO 170
240 & XBOX(X,Y) AT 2,2
250 X= X-7: Y= Y-3
260 GOTO 230
```

14

Fantavision

What Is Fantavision?

Fantavision is an extremely versatile and effective animation software package published by Broderbund Software. It consists of a system disk (which permits the making of one backup copy) and a user's manual. A collection of movie demonstrations and sample background scenes is located on the back side of the system disk. You can store animation created with Fantavision on a separate disk called a data or program disk.

A unique feature of Fantavision is that it provides a system in which the intermediate images between the start and finish of any object movement are generated automatically. This feature, called "tweening," eliminates most of the time-consuming and onerous drawing so characteristic of other animation systems. Employing a related capability, Fantavision can also transform one shape into another by automatically producing a series of transitional forms.

An Important Note

Fantavision is designed for use in the single high-resolution mode. It will not function with previously stored double high-resolution pictures. Let us hope that a double HiRes version will be available in the future.

What's Needed for Fantavision?

You need the following components to run Fantavision:

Apple II series computer with 64K
Disk drive
Color TV or monitor
Mouse, Apple Graphics Tablet, Koala Pad, or joystick

Movies a la Fantavision

As you might reasonably expect, the main use of Fantavision is to make movies. The quality of these movies is good enough to be used for anything from games to business applications or low-cost TV commercials.

Like their photographic counterparts, Fantavision movies are made up of frame sequences in which positional changes of objects create the illusion of motion.

Fantavision movies are made with or without stationary background scenes against which objects appear to move. You can create these backgrounds by using Fantavision's drawing features or by loading previously stored single HiRes pictures. In some cases, backgrounds are not needed, but in general, they add a great deal to the presentation of a story.

The Fantavision Concept

Just in case you haven't seen Fantavision in action at your favorite software emporium, let's run through a conceptual description of how it's used in making a very simple movie. This movie will show a vertical stick falling to a horizontal position. (In this example let's not worry about how lines are drawn, frames are counted, etc.—we'll get to that later.)

First, we'll draw a vertical line at the left of the screen and call this picture frame #1. Then we'll draw a horizontal line at the right side of the screen and call that frame #2. This completes our part of the moviemaking procedure. (Yes, we just made a two-frame movie!) Now let's show it.

When we tell the computer to run the movie, the line at the left will appear to tip over and fall down to the horizontal position at the right with a number of visible intermediate positions in between.

We drew only two lines. Where did all those intermediate images come from? Fantavision and your Apple computer automatically calculated and drew them! This amazing capability of Fantavision to generate changes in the size and position of objects has previously been available only in equipment and software costing thousands of dollars.

Now that you have the key to the Fantavision concept, let's take a look at the long list of features offered by this cleverly designed animation package.

Fantavision Features

Here's a partial list of features you can use in creating Fantavision images:

1. Select colors.
2. Draw with a series of points.
3. Operate in a Zoom mode.
4. Insert or delete points.
5. Create dots of various sizes.
6. Place text on the screen.
7. Make rectangles.
8. Make circles.

Fantavision also has many features used in the animation process:

1. Frame counter
2. Object counter
3. Load, save, and clear Movie
4. Load, save, and clear Backdrop
5. Normal, background, lightning, and trace mode animation
6. Shape transformation
7. Animation speed setting
8. Frame zap
9. Ability to turn, lean, flip, or flatten objects
10. Ability to transform objects from one shape to another
11. GO command (to run a movie)

When you "boot up" Fantavision, you'll see all of these features on the screen as shown in Figure 14.1. You can control many of Fantavision's features from the Apple keyboard, but chances are that most users will be using the mouse (or other pointing device) in preference to the keyboard.

A Fantavision Project

With so many sophisticated features at your fingertips, Fantavision makes you feel like a kid with a pocketful of nickels in a candy store (before inflation!). When in doubt about where to start, "Keep it simple" is the best answer. We'll make a simple little movie to familiarize you with the basics of Fantavision. This movie will show a glass of bubbling liquid (wine, perhaps?).

AT-A-GLANCE

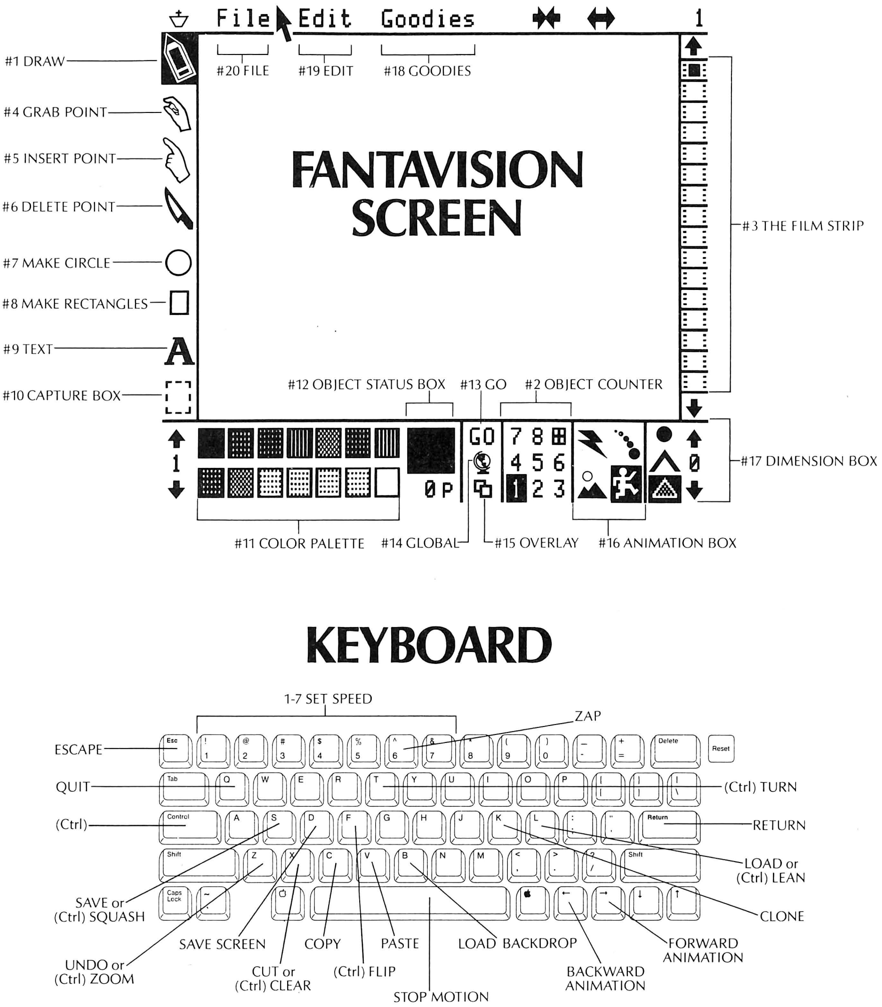

File
Edit
Goodies
1
#1 DRAW
#20 FILE
#19 EDIT
#18 GOODIES
#4 GRAB POINT
#5 INSERT POINT
#6 DELETE POINT
#7 MAKE CIRCLE
#8 MAKE RECTANGLES
#9 TEXT
#10 CAPTURE BOX
FANTAVISION SCREEN
#3 THE FILM STRIP
#12 OBJECT STATUS BOX
#13 GO
#2 OBJECT COUNTER
GO
7 8
4 5 6
1 2 3
0 P
0
#17 DIMENSION BOX
#11 COLOR PALETTE
#14 GLOBAL
#15 OVERLAY
#16 ANIMATION BOX
KEYBOARD
1-7 SET SPEED
ZAP
ESCAPE
QUIT
(Ctrl)
(Ctrl) TURN
RETURN
LOAD or (Ctrl) LEAN
CLONE
SAVE or (Ctrl) SQUASH
UNDO or (Ctrl) ZOOM
SAVE SCREEN
COPY
PASTE
LOAD BACKDROP
FORWARD ANIMATION
BACKWARD ANIMATION
CUT or (Ctrl) CLEAR
(Ctrl) FLIP
STOP MOTION

Figure 14.1 ***The Fantavision Screen and Keyboard***

Starting as small bubbles at the bottom of the glass, the bubbles will increase in size as they rise to the liquid surface. This process will continue until we stop the movie.

Here's how to put "bubbles in the wine":

1. Click your mouse over the pencil icon at the top left of the screen to select "Draw."
2. Choose black in the color palette.
3. Select a line drawing mode by clicking the open hash mark at the lower right of the screen.
4. Draw a large "glass of wine" using the mouse to set points that are joined by straight lines.
5. Pull down the File menu and select "Save Backdrop." Answer the prompt request for a name with "Glass" (no quotes). Press Return.
6. Pull down the File Menu and select "Clear Movie."
7. Pull down the File menu and select "Load Backdrop." Select the name "Glass." The frame counter in the upper right corner will now be set at 1.
8. Now select another color from the color palette.
9. To make bubbles, select the dot above the hash mark for drawing.
10. Set the dot size to 1 by clicking the arrows at the extreme lower right of the screen.
11. Use the mouse to place several small dots at the bottom of the glass. Click the arrow at the bottom of the column of movie frames (right of screen) to set the frame number to 2.
12. Repeat this process, increasing the dot (bubble) size. Raise the dot position as you increase the size. Remember to give each new drawing a frame number.

13. After drawing bubbles at the surface of the wine, repeat the whole process in reverse. Remember to change the frame number for each picture situation. You'll use about 15 frames for the entire action.
14. Now pull down the File menu and select "Save Movie." Name it "Bubbles." Press Return.
15. Click the mouse over GO and watch your movie run. Amazing, isn't it?
16. To stop the movie at any point, just press Return. Select "GO" to restart it.
17. To clear the screen for further work, pull down the File menu and select "Clear Movie."
18. To run any movie with a background, remember that you must load the movie *and* the background before giving the GO command. If you don't load the background, it won't be there.

Looking Ahead

This example of how Fantavision works barely scratches the surface of its possibilities. An hour spent in hands-on experience with this truly excellent animation package will fill you with enthusiasm for its great capabilities. Fantavision offers the creative user a means of enhancing his or her talents in ways never before achievable with a home computer software package.

Appendix

More Demonstration Programs

For the Finale

Here are some additional programs to give you more fun with Apple graphics. Some are fast, some are slow, and the images they create vary enormously. Be sure to try all of them! The "Program Notes" preceding each listing will give you some idea of what to expect.

Speed and Color

Program A1.1, written in low-resolution graphics form, creates a complex structure of linearity and color. It is a dynamic program that gives viewers the impression of a three-dimensional display.

Try running this program with a Lionel Hampton accompaniment!

Program A1.1 **Fastrack Colors**

```
1  REM PROGRAM A1.1 FASTRACK COLORS
10  HOME : GR : CALL - 1998
20  POKE 49234,0
30  COLOR= RND (1) * 16
40  Y= RND (1) * 47
50  HLIN 0,39 AT Y
60  X= RND (1) * 39
70  VLIN 0,47 AT X
80  GOTO 30
```

The Last Sunset

Program A1.2 offers a symbolic sunset that could be just what you need as a final slide or closing scene in a video presentation. For some novel effects on your screen, try changing the color values used in this program.

Program A1.2 **Sunburst**

```
1  REM PROGRAM A1.2 SUNBURST
10  GOTO 5000
20  HGR : POKE 49234,0 : HCOLOR= 2
30  X1 = 0 : FOR X= 0 TO 279 STEP 20
40  Y1 = 0 : FOR Y= 0 TO 189 STEP 15
50  HPLOT X1, Y1 TO X, Y
60  NEXT Y : NEXT X
70  HCOLOR= 5
80  FOR X= 0 TO 279 STEP 15
90  FOR Y= 0 TO 189 STEP 15
100  HPLOT X1, Y1 TO X, Y
110  NEXT Y : NEXT X
120  HCOLOR= 3 : SCALE= 15
130  FOR R= 0 TO 64 : ROT= R
140  DRAW 1 AT 18, 15 : NEXT R
150  GOTO 120
```

```
5000  POKE 232,252 : POKE 233,29
5010  FOR LOC= 7676 TO 7682
5020  READ BYTE : POKE LOC, BYTE
5030  DATA 1, 0, 4, 0, 15, 5, 0
5040  NEXT LOC
5050  GOTO 20
```

Copycat Styling

Program A1.3 demonstrates how the linearity of paintings in the style of Piet Mondrian can be replicated by simple programming.

For much faster operation and a greater color selection, convert this program to the low-resolution mode.

Program A1.3 **Mondrianish**

```
1  REM PROGRAM A1.3 MONDRIANISH
10  HOME : HGR
20  HCOLOR= 3 : HPLOT 0, 0
30  CALL 62454 : HCOLOR= 4
40  REM LEFT EDGE
50  FOR X= 0 TO 20
60  FOR Y= 0 TO 180
70  HPLOT X, Y : NEXT Y : NEXT X
80  REM NEXT VERT. STRIPE
90  FOR X1 = 33 TO 36
100  FOR Y1 = 0 TO 180
110  HPLOT X1, Y1
120  NEXT Y1 : NEXT X1
130  REM BLUE PANEL
140  HCOLOR= 2 : FOR X2= 21 TO 32
150  FOR Y2= 108 TO 180
160  HPLOT X2, Y2 : NEXT Y2
170  NEXT X2
180  REM NEXT VERT. STRIPE
190  HCOLOR= 4
200  FOR X3= 83 TO 86
```

(continued on the next page)

```
210 FOR Y3= 0 TO 180 : HPLOT X3, Y3
220 NEXT Y3 : NEXT X3
230 REM RT. BORDER
240 FOR X4= 269 TO 279
250 FOR Y4= 0 TO 180
260 HPLOT X4, Y4 : NEXT Y4 : NEXT X4
270 REM SHORT VERT. STRIPE
280 FOR X5= 245 TO 248
290 FOR Y5= 108 TO 180
300 HPLOT X5, Y5 : NEXT Y5 : NEXT X5
310 REM RED BLOCK
320 HCOLOR= 5
330 FOR X6= 249 TO 268
340 FOR Y6= 108 TO 180
350 HPLOT X6, Y6
360 NEXT Y6 : NEXT X6
370 REM HORIZ. LINE
380 HCOLOR= 4 : FOR X7= 0 TO 279
390 FOR Y7= 105 TO 108
400 HPLOT X7, Y7
410 NEXT Y7 : NEXT X7
```

Strike Up the Band!

Program A1.4 was written to create imagery for my experimental film *Musique Graphique.* I urge you to run this program with an accompaniment of "The Stars and Stripes Forever" played by a good military band! You'll quickly decide which measures of the music should be "synched" to the descending triangles on screen.

The experience of adding music to just one program will send bundles of neurons into the creative side of your brain! You'll think of music that fits other programs—it's a whole new world! *Please* try it.

Author's note: In my opinion, it is almost impossible to exaggerate the value of adding rhythmically related music to dynamic computer displays.

Program A1.4 **Musique Triangles**

```
1  REM PROGRAM A1.4 MUSIQUE TRIANGLES
10  GOTO 5000
20  HGR : POKE 49234,0
30  FOR R= 0 TO 64 : ROT= R : SCALE= 35
40  A= INT ( RND ( 1 ) * 8 )
50  IF A= 0 OR A= 4 THEN A= 5
60  HCOLOR= A : DRAW 1 AT 140, 100
70  NEXT R : V= 1 : X= 140
80  HCOLOR= INT ( RND ( 1 ) * 8 )
90  FOR Y= 0 TO 180
100  SCALE= V : DRAW 1 AT X, Y
110  V= V + .5 : NEXT Y : HCOLOR= 2
120  HPLOT 140, 0 TO 245, 150
130  HPLOT 245, 150 TO 231, 180
140  HCOLOR= 5
150  HPLOT 140, 0 TO 50, 180
160  HPLOT 50, 180 TO 232, 180
170  HPLOT 141, 0 TO 232, 180
180  GOTO 30
5000  POKE 232,252 : POKE 233,29
5010  FOR LOC= 7676 TO 7682
5020  READ BYTE : POKE LOC, BYTE
5030  DATA 1, 0, 4, 0, 18, 63, 0
5040  NEXT LOC
5050  GOTO 2
```

Coming Abstractions

Program A1.5 features the creation of triangular shapes bearing abstract patterns in a variety of colors. To put some "wow" in your color patterns, try varying the ROT values used in this program.

Program A1.5 **Triangular Abstractions**

```
1 REM PROGRAM A1.5 TRIANGULAR ABS.
10 GOTO 1000
20 HGR : POKE 49234,0
30 A= INT ( RND ( 1 ) * 8 )
40 IF A < 1 OR A= 4 THEN A= 5
50 HCOLOR= A
60 X= INT ( RND ( 1 ) * 280 )
70 Y= INT ( RND ( 1 ) * 189 )
80 ROT= 15
90 FOR Q= 1 TO 40 : SCALE= Q
100 DRAW 1 AT X, Y
110 NEXT Q
120 GOTO 30
1000 POKE 232,252 : POKE 233,29
1010 FOR LOC= 7676 TO 7682
1020 READ BYTE : POKE LOC, BYTE
1030 DATA 1, 0, 4, 0, 18, 63, 0
1040 NEXT LOC
1050 GOTO 20
```

Cascading Color

An intriguing cascade of colors is formed by program A1.6. For some fascinating fun, try changing the shape data (between the second and third 0's) in line 1030. If you change the number of bytes of shape data, don't forget to make a corresponding change in line 1010.

Program A1.6 **Color Cascade**

```
1 REM PROGRAM A1.6 COLOR CASCADE
10 GOTO 1000
20 HGR : POKE 49234,0
30 A= INT( RND (1) * 8)
40 IF A < 1 OR A= 4 THEN A= 5
50 HCOLOR= A : ROT= 48
```

```
60  X= INT( RND (1) * 279 : Y= 0
70  FOR Q= 1 TO 50 : SCALE= Q
80  DRAW 1 AT X, Y : NEXT Q
90  GOTO 30
1000  POKE 232,252 : POKE 233,29
1010  FOR LOC= 7676 TO 7686
1020  READ BYTE : POKE LOC, BYTE
1030  DATA 1,0,4,0,18,196,63,53,255,196,0
1040  NEXT LOC
1050  GOTO 20
```

Computempo, Anyone?

Whether your musical taste runs to rock, Bach, country, or concerti, you'll enjoy running the next two programs with the accompaniment of your choice.

Notice that all of the screen action is upward in Program A1.7, while some color bars go up and some go down in Program A1.8. Try a bit of Jean-Pierre Rampal's flute magic with the latter program!

A little experimentation with a variety of rhythms will give you the urge to add a musical accompaniment to all of your dynamic programs.

Program A1.7 **Computempo**

```
1  REM PROGRAM A1.7 COMPUTEMPO
10  HOME : GR : POKE 49234,0
20  CALL - 1998
30  C= RND (1) * 16
40  IF C= 0 THEN C= 13
50  COLOR= C
60  X= RND (1) * 39
70  VLIN 0, 47 AT X
80  FOR Y1 = 39 TO 0 STEP - 1
90  COLOR= 0
100  HLIN 0, 39 AT Y1 : NEXT Y1
110  GOTO 30
```

Program A1.8 **Tempo Dots**

```
1  REM PROGRAM A1.8 TEMPO DOTS
10  HOME : GR : POKE 49234,0
20  CALL – 1998 : A= 39
30  B= 1 – B
40  C= RND (1) * 16
50  IF C= 0 THEN C= 13
60  COLOR= C
70  X= RND (1) * 39
80  VLIN 0, 47 AT X
90  IF Q > 0 THEN GOTO 150
100  FOR Y1 = 0 TO A
110  COLOR= 0
120  HLIN 0, RND (1) * 39 AT Y1
130  NEXT Y1
140  Q= 1 – B : GOTO 30
150  FOR Y1 = A TO 0 STEP – 1
160  GOTO 110
```

In the Best Circles

Program A1.9 demonstrates what happens when you draw square shapes at X,Y coordinates derived from a modified circle program. To "fatten" the circle of squares created by this program, increase the value of the multiplier in line 30.

Program A1.9 **Shapecircle**

```
1  REM PROGRAM A1.9 SHAPECIRCLE
10  GOTO 5000
20  HGR : POKE 49234,0 : R= 60
30  FOR I= 0 TO 6.28 STEP .15
40  X= 1.2 * R * SIN (I)
50  Y= R * COS (I)
60  ROT= 0 : SCALE= 20 : HCOLOR= 6
```

```
70  DRAW 1 AT 140 + X, 80 + Y
80  NEXT I : GOTO 30
5000  POKE 232,252 : POKE 233,29
5010  FOR LOC= 7676 TO 7682
5020  READ BYTE : POKE LOC, BYTE
5030  DATA 1,0,4,0,44,62,0
5040  NEXT LOC
5050  GOTO 20
```

Index

A$2000, 66
A$4000, 66, 68
Address, Shape Data, 36
Ampersand commands, 97, 156
Animation
  general, 55
  with Fantavision, 157-163
Auxiliary memory in Beagle Graphics (.Aux), 98

Backups, 5
Beagle Graphics software
  commands, 97
  demonstration project, 133-138
  described, 129-131
  printing Beagle Graphics, 130
BLOAD, 66, 150
& BOX, 154
BRUN, 148
BSAVE, 66
CALL 768, 145, 150, 151
CALL 62454, 53
CALL – 1998, 7, 14
CALL – 958, 63
Camera
  exposures, 76
  types, 71-73
& CIRCLE, 155
Circle program, 29
Color
  GR calibration, 15
  GR color commands, 12
  GR color list, 12
  HGR color list, 24
  screen color in double HiRes, 151
Conceptual programming
  demonstration programs, 58, 59, 61
  introduction to, 21
Conversion programs
  Beagle Graphics pictures to Dazzle Draw, 141

  - Dazzle Draw pictures to Beagle Graphics, 142
  - Dazzle Draw pictures to Complete Graphics System, 143
  - DOS 3.3 picture to ProDOS, 140
  - HGR program to DHGR, 150-153
  - HiRes picture to double HiRes, 145
- Curved lines
  - HiRes programs, 28
  - LoRes programs, 20

- Data disks
  - details of, 94
  - for DOS 3.3 system, 95
  - for ProDOS system, 95
- Dazzle Draw software
  - demonstration pictures, 107
  - demonstration project, 104-107
  - described, 99, 100
- DHGR utility, 150, 153
- DOS 3.3 overview, 3
- Dots, Random Dots program, 16
- Double HiRes
  - discussion of, 96, 98, 147
  - programs, 154, 155
- DRAW, 34

- Fantavision software
  - demonstration project, 160-163
  - described, 157-161
- Font disks, 92
- Fontrix software
  - demonstration project, 113-118
  - described, 109
  - font demonstration, 111
  - Fontpak disks, 112
  - font styles, 110
  - printing Fontrix, 109
- FOR—NEXT, 7

- & HGR, & HGR2, 152
- HGR, HGR2, 24
- HiRes graphics, 23
  - commands demonstrated, 25-31
- HLIN, VLIN, GR commands, 7, 12
- HPLOT, HGR command, 25
- HPLOT TO, HGR command, 26
- HTAB, VTAB, 13, 63

- LoRes graphics, 11

- Menu-driven system defined, 94
- Music, adding sound to video, 84, 85

- PEEK (=16336), 63
- Photography
  - camera types, 71-73
  - exposure times, 76
  - film selection, 76
  - ground rules, 70
  - techniques, 72-76
  - warm tone filter, 76

PLOT, GR command, 12
POKE 49234,0, 14
Print Shop, The
  demonstration project, 123-127
  described, 119-121
  hard copy printout, 126
ProDOS overview, 4
Programming,
  DHGR, 154
  GR, 12-20
  HGR, 24-31

Random numbers, 8
ROT, HGR command, 34

SCALE, HGR command, 7, 34
Shape
  data for 5 shapes, 45
  demonstration programs, 37, 40, 42, 43, 48-53
  examples, 45
  multiple shapes in one table, 46
  shape commands, 34
Shape table rules, 35, 46
Shape test programs
  rotating test, 43
  Shape Tester, the, 42
Signs and sequence, 7
Slide show
  discussion, 65-85
  program, 68
Startup
  disks, 96
  programs, 149
Symtec
  mixing video and computer graphics, 86
System disks
  configuring, 92
  introduction, 91

Text, 13, 24
Traveling text, 63

Video
  audio dubbing, 85
  camera action, 83
  choice of VCR, 78
  computer video, 80
  connections, 79
  illumination problems, 82
  practical instructions, 80

Windows
  Dazzle Draw, 102
  defined, 94

XDRAW, 35

Computers $16.95

Put the most advanced color graphics at your fingertips with…

HiRes/Double HiRes GRAPHICS

For the Apple IIc and Apple II Family

The newest generation of high resolution graphics techniques can produce images of startling sharpness and brilliance, with more colors than high resolution (HiRes) and double the definition. *HiRes/Double HiRes Graphics* hands you all the tools you need to paint and draw advanced graphics with available software packages or your own programs. You'll be able to create, store, recall, and draw HiRes and double HiRes graphics including dynamic ones that change in color and configuration as their programs run.

All the necessary programming and operational details are here, with explanations of low and high resolution commands, a discussion of shapes and shape tables, and dozens of demonstration programs.

DeWitt's guide also explains a variety of techniques for generating graphics by custom modifying off-the-shelf graphics packages, and by using peripheral devices and ProDOS commands. The capabilities of popular software packages such as Dazzle Draw, The Print Shop, Fontrix, Beagle Graphics, and Fantavision are explored and their typical projects described. Sample sections show you how to use the software surveyed, and color screen illustrations provide a glimpse of the graphics generated by each package. And DeWitt presents instructions on photographing and videotaping images directly from your computer monitor using conventional cameras and video recorders.

With its clear explanations and complete instructions, *HiRes/Double HiRes Graphics* will open up a colorful new world of possibilities in computer graphics for your business, classroom, design, or computer needs.

William H. DeWitt is President of Computer Art. He has taught computer graphics at the University of Rochester and is a Fellow of the Institute of Fellows, Rochester Institute of Technology. He is the author of *Art and Graphics on the Apple II/IIe* (Wiley Press).

Wiley Press guides have taught more than three million people to use, program, and enjoy microcomputers. Look for them all at your favorite bookshop or computer store.

WILEY PRESS
a division of JOHN WILEY & SONS, Inc.
Business/Law/General Books Division
605 Third Avenue, New York, N.Y. 10158
New York • Chichester • Brisbane •
Toronto • Singapore

ISBN 0 471-83183-2

DeWitt

HiRes/Double HiRes GRAPHICS

UR-375-789 *